Life Language Processing: Deep Learning-based Language-agnostic Processing of Proteomics, Genomics/Metagenomics, and Human Languages

by

Ehsaneddin Asgari

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Applied Science and Technology

and the Designated Emphasis

in

Computational Data Science and Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Mohammad R.K. Mofrad, Chair
Professor Shaofan Li
Assistant Professor David Bamman

Summer 2019

Life Language Processing: Deep Learning-based Language-agnostic Processing of Proteomics, Genomics/Metagenomics, and Human Languages

Abstract

Life Language Processing: Deep Learning-based Language-agnostic Processing of Proteomics, Genomics/Metagenomics, and Human Languages

by

Ehsaneddin Asgari

Doctor of Philosophy in Applied Science and Technology

and the Designated Emphasis

in

Computational Data Science and Engineering

University of California, Berkeley

Professor Mohammad R.K. Mofrad, Chair

A broad and simple definition of 'language' is a set of sequences constructed from a finite set of symbols. By this definition, biological sequences, human languages, and many sequential phenomena that exist in the world can be viewed as languages. Although this definition is simple, it includes languages employing very complicated grammars in the creation of their sequences of symbols. Examples are biophysical principles governing biological sequences (e.g., DNA, RNA, and protein sequences), as well as grammars of human languages determining the structure of clauses and sentences. This dissertation uses a language-agnostic point of view in the processing of both biological sequences and human languages. Two main strategies are adopted toward this purpose, (i) character-level, or more accurately, subsequence-level processing of languages, which allows for simple modeling of the sequence similarities based on local information or, bag-of-subsequences, (ii) language model based representation learning encoding contextual information of sequence elements using the neural network language models. I propose language-agnostic and subsequence-based language processing using the above-mentioned strategies in addressing three main research problems in proteomics, genomics/metagenomics, and natural languages using the same point-of-view.

One of the main challenges in proteomics is that there exists a large gap between the number of known protein sequences and known protein structures/functions. The central question here is how to efficiently use large numbers of sequences to achieve a better performance in the structural and functional annotation of protein sequences. Here, we proposed subsequence-based representations of protein sequences and their language model-based embeddings trained over a large dataset of protein sequences, which we called protein vectors (or ProtVec). In addition, we introduced a motif discovery approach, benefiting from probabilistic segmentation of protein sequences to find functional and structural motifs. This

segmentation is also inferred from large protein sequence datasets. The ProtVec approach has proved a seminal contribution in protein informatics and now is widely used for machine learning based protein structure and function annotations. We showed in different protein informatics tasks that bag-of-subsequences and protein embeddings are complementary information for language-agnostic prediction of protein structures and functions, which also achieved the state-of-the-art performance in the 2 out of 3 tasks of Critical Assessment of protein Function Annotation (CAFA) in 2018 (CAFA 3.14). Moreover, we systematically investigated the role of representation and deep learning architecture in protein secondary structure prediction from the primary sequence. Publicly available tools are provided for achieving state-of-the-art performance accuracy that can be further expanded by the community.

One of the prominent challenges in metagenomics involves the host phenotypic characterization based on the associated microbial samples. Microbial communities exist almost on every accessible surface on earth, supporting, regulating, and even causing unwanted conditions (e.g., diseases) to their hosts and environments. Detection of the host phenotype and the phenotype-specific taxa from the microbial samples is the chief goal here. For instance, identifying distinctive taxa for microbiome-related diseases is considered key to the establishment of diagnosis and therapy options in precision medicine and imposes high demands on the accuracy of microbiome analysis techniques. Here, we propose two distinct language-agnostic subsequence-based processing methods for machine learning on 16S rRNA sequencing, currently the most cost-effective approach for sequencing of microbial communities. We propose alignment- and reference- free methods, called MicroPheno and DiTaxa, designed for microbial phenotype and biomarker detection, respectively. MicroPheno is a k-mer based approach achieving the state-of-the-art performance in the host phenotype prediction from 16S rRNA outperforming conventional OTU features. DiTaxa, substitutes standard OTU-clustering by segmenting 16S rRNA reads into the most frequent variable-length subsequences. We compared the performance of DiTaxa to the state-of-the-art methods in phenotype and biomarker detection, using human-associated 16S rRNA samples for periodontal disease, rheumatoid arthritis, and inflammatory bowel diseases, as well as a synthetic benchmark dataset. DiTaxa performed competitively to MicroPheno (state-of-the-art approach) in phenotype prediction while outperforming the OTU-based state-of-the-art approach in finding biomarkers in both resolution and coverage evaluated over known links from literature and synthetic benchmark datasets.

The third central problem we addressed in this dissertation is focused on human languages. Many of 7000 world's natural languages are low-resource and lack digitized linguistic resources. This has put many of these human languages in danger of extinction and has motivated developing methods for automatic creation of linguistic resources and linguistic knowledge for low-resource languages. To address this problem via our language-agnostic point of view (by not treating different languages differently), we develop SuperPivot for subsequence-based linguist marker detection in parallel corpora of 1000 languages, which was the first computational investigation for linguistic resource creation in such a scale. As an example, SuperPivot was used to study the typology of tense in 1000 languages. Next, we utilized

SuperPivot for the creation of the largest sentiment lexicon to date in terms of the number of covered languages (1000+ languages) achieving macro-F1 over 0.75 on word sentiment prediction for most evaluated languages, meaning that we enable sentiment analysis in many low resource languages. To ensure the usability of *UniSent* lexica for any new domain, we propose *DomDrift*, a method quantifying the semantic changes of words in the sentiment lexicon in the new domain. Next, we extend the *DomDrift* method to quantifying the semantic changes of all words in the language. We proposed a new metric for language comparisons based on the language word embedding graphs requiring only monolingual embeddings and word mapping between languages obtained through statistical alignment in parallel corpora. We performed language comparison for fifty natural languages and twelve genetic language variations of different organisms. As a result, natural languages of the same family were clustered together. In addition, applying the same method on organisms' genomes confirmed a high-level difference in the genetic language model of humans/animals versus plants. This method called word embedding language divergence is a step toward unsupervised or minimally supervised comparison of languages in their broad definition.

In dedication to
the kindest and the most compassionate mother, **Fatimah al-Zahra** (M.I.b.S.f.H),
from whom, her father, her husband, and her family I have everything and have nothing without them.

I do not deserve it, but she blesses me with the greatest gifts and the sources of mercy in my life, among them: (i) the love of my life, Meshkat, who inspires me, supports me, encourages me, and has been bear with me in every single moment of life ever since I was a first-year PhD student, (ii) my beloved parents whom I could never compensate their efforts for me and my dear parents-in-law. They raised us with their unconditional love and support, and tolerated the far physical distance between us at no complaint. I eagerly bend my knees kissing their hands out of deepest respect and love. (iii) and other teachers/friends who are enlightening what is dark in me. Thank you, my God, for all, all praise belongs to you, Lord of the worlds, and I solely rely upon you.

Contents

List of Figures

List of Tables

Nomenclature

16S rRNA gene: The 16S rRNA gene is a highly conserved gene across bacteria and archaea allowing for differential identification of *taxon* identities and relative abundances.

Accuracy metric Accuracy is an evaluation metric of the *classification* models defined as the number of correct predictions over the total number of predictions.

Alignment (sequence) Sequence alignment in bioinformatics refers to vertical aligning of similar subsequences of 2 or more biological sequences (DNA, RNA, or protein) to immediately identify the similarities that are related functionally, structurally, or evolutionary.

Alignment (word) A bi-part graph relating textual units (words) of a sentence in the *source language* to its translation in the *target language*

Amino acid Monomers of ≈ 20 types, which are the elements for the creation of protein polymers

Annotation Annotation in datascience refers to assigning metadata to data, which can be done *manually* or automatically

Biomarker "A biomarker is a biological characteristic that is objectively measured and evaluated as an indicator of normal biological or pathological processes or a response to a therapeutic intervention." (*Nature.com* definitions)

Bootstrapping Bootstrapping refers to performing measurments using random *sampling* with replacement in statistical analysis.

Classification Classification is the process of categorizing elements according to their observed variables. In machine learning elements are data instances, and the observed variables are data *representations*. In biology, the elements are different organisms, the observed variables are their various characteristics in their genome or their behaviors, and the categories are the *taxonomic groups*.

Classifier In machine learning, classifier is a mathematical function mapping the given data points from the input representation space to the *class/label* representation space.

Class In machine learning, 'class' refers to the part of the metadata identifying the category of a given data instance. The automatic prediction of the 'class' is the goal of the *classification task.*

Clustering Clustering refers to the process of grouping data points according to their observed variables. The difference between clustering and classification is that, in clustering, these groups are not pre-defined (clustering is *unsupervised* and $classification$ is *supervised*).

CNN See Convolutional neural network

Convolutional Neural Network Convolutional neural network (or CNN) is a category of *deep learning models* that is very common in machine learning analysis of images/videos. A *convolutional* layer applies trainable *convolutional* functions on its input.

Convolution *Convolution* is a mathematical operation between two function f and g, that can be defined as $(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau$, or in a discrete case $(f * g)[k] = \sum_{n=-\infty}^{\infty} f[n]g[k-n]$.

Corpus A collection of written texts usually sharing some characteristics (e.g., being in a specific language, or being from particular authors, genres, etc.)

Crohn's disease "Crohn's disease is a chronic inflammatory disease of the gastrointestinal tract, mostly affecting the ileum. The inflammation can extend through the intestinal wall and tends to be asymmetric and in patches, with granulomas forming in some patients. Genetic, environmental, immunological, and bacterial factors are all thought to contribute to the disease." (*Nature.com* definitions)

Dataset Dataset refers to a collection of data points usually coming together with the metadata information in table structure.

Deep learnig Deep learning is a family of *machine learning* methods based on neural networks, where the learning can be *supervised*, *semi-supervised* or *supervised.*

Disordered proteins A type of, or a region in *proteins* lacking a fixed or ordered 3D structure.

DisProt The *dataset* of *disordered protein* sequences

Distance In data analysis, distance refers to the mathematical measurement indicating how far are two data points. Different data types (e.g., probabilities, real vectors, etc.) requires their proper measures of distance. Euclidean distance, KL divergence, mean squared error are instances of the defined distances between two data objects.

DNA “DNA (deoxyribonucleic acid) is the nucleic acid polymer that forms the genetic code for a cell or virus. Most DNA molecules consist of two polymers (double-stranded) of four nucleotides (A, T, C, G) that each consist of a nucleobase, the carbohydrate deoxyribose, and a phosphate group, where the carbohydrate and phosphate make up the backbone of the polymer. (*Nature.com* definitions)

Domain (language) domain refers to a specific language setting or context sharing a set of meanings (e.g., sports domain or law domain).

Dropout (neural network) Dropout is a regularization method in neural network training, which reduces the overfitting by dropping hidden units.

Embedding word embedding is a popular vector representation of words in the machine, in a lower dimension than the vocabulary size of *corpus* (usually less than 1000). This representation is automatically learned using a large collection of texts and facilitates measurement of words similarities through vector similarities.

Environment Environment, in general, refers to the surroundings. In microbiology, environments can refer to the water, soil, sediment, plant, or even animals and humans, where the bacteria can exist.

Evaluation In *machine learning*, evaluation refers to the process of testing the machine learning model against expected outputs/behaviours.

F1-score F1 is a metric for the *evaluation* of machine learning $classification$. F1 is defined as a harmonic average of *precision* and $recall$. *Micro* and *macro* averaged F1s can be reported (see *micrometrics* and *macrometrics*)

Family A language family refers to a group of languages having a common ancestral language. In the biology domain, similarly, a family is a *taxonomic* rank.

Feature In machine learning, feature refers to a measurable property or characteristic of a data instance that is observed and can be represented as vectors or matrices for the machine.

Gene Gene refers to the functional sub-sequences of *nucleotides* in *DNA* or *RNA*.

Graph In mathematics, graph is an abstraction of data containing a set of vertices and edges connecting them. The edges can be weighted or unweighted, directed or undirected

Integrin Integrins are transmembrane receptor proteins with a critical role in the cell adhesion

k-mer In bioinformatics, k-mers are sub-sequences of length k within a biological sequence (*DNA, RNA, protein*). In the language processing domain, $k - mers$ are called character *n-grams*.

KL divergence The Kullback–Leibler divergence (or shortly KL divergence) is a measure of *distance* between two probability distributions.

Label In machine learning, the *label* usually refers to the output of $classification$, i.e., the category of data.

Language-agnostic Independent of any specific language or prior assumption about it.

Language "a (finite or infinite) set of sentences, each finite in length and constructed out of a finite set of elements" (Chomsky 2002)

Layer (deep learning) A *deeplearning* model is usually a hierarchical model of non-linear transformations, each of these transformations is called a layer.

Lexica A plural form of $lexicon$.

Lexicon Lexicon can be defined as a textual *dataset*, containing information about words and their different categories.

Machine Learning Machine learning is a computer science field of study enabling the machines to make predictions/decisions without following any explicitly given instruction set. Most of machine learning approaches infer their prediction models through given examples in a mathematical or statistical framework.

Macro metrics Macro-metrics are a type of $evaluation$ metrics for $multi-class\ classification$, which takes the unweighted average of metrics for the individual $classes$. Thus, independent to the class priors, we give equal importance to all $classes$ in the $evaluation$.

Marker (linguistic) A marker is an easily distinguishable text unit indicating a specific grammatical function

Metagenome Metagenome refers to the collection of the genetic material belonging to both the host and the microorganisms (*microbiome*) living at an environemnt

Micro metrics Micro-metrics are a type of $evaluation$ metrics for $multi-class\ classification$, which takes the weighted average of metrics over all $classes$. Thus, $classes$ which are more likely have greater influence in the $evaluation$.

Microbiome "The microbiome comprises all of the genetic material within a microbiota (the entire collection of microorganisms in a specific niche, such as the human gut). This can also be referred to as the metagenome of the microbiota." (*Nature.com* definitions)

Morpheme A morpheme is the smallest unit of text with meaning (dependent or freestanding) that cannot be further divided.

Motif $Protein$ short linear motif (SLiM) sequences are short sub-sequences of usually 3 to 20 $aminoacids$ that are presumed to have important biological functions; examples of such patterns are cleavage sites, degradation sites, docking sites, ligand binding sites, etc.

Multi-class (classification) Multi-class $classification$ refers to a $machinelearning\ classification$ setting, where the model has to choose between a set of possible categories.

n-gram Refer to the definition of $k - mer$.

Neural network Neural networks, or more accurately Artificial Neural Networks, are computational models or interconnected $layers$ inspired from the human neural network to perform specific prediction tasks.

Nucleotides Nucleotides are the basic units of nucleic acids such as DNA or RNA.

OTU Operational Taxonomic Unit or shortly OTUs are $clusters$ of similar sequence variants of the $16SrRNA$. OTUs are used to classify groups of closely related sequences.

Parallel corpus A parallel corpus is a collection of texts, where each sentence in it is translated into one or more other languages

Periodontal disease Periodontal disease is an inflammatory disease caused by specific microorganisms in supporting tissues of teeth.

Phenotype Phenotype refers to the observable characteristics or traits of organisms.

Pipeline Pipeline is a sequence of data processing modules, where the output of a module is the input to the next one, and they are connected to accomplish a particular task.

Precision In a machine learning $classification$ task, precision for a class is defined as the number of true positives divided by the total number of elements labeled as belonging to the class. $Micro$ and $macro$ averaged precisions can be reported (see $micrometrics$ and $macrometrics$).

Primary sequence (protein) The primary sequence/structure of the protein is the linear sequence of amino acids, which is also known as the protein sequence.

Protein Proteins are macromolecules (polymers) consisting of the chain(s) of smaller elements (monomers) called *amino acids*.

Proteomics "Proteomic analysis (proteomics) refers to the systematic identification and quantification of the complete complement of proteins (the proteome) of a biological system (cell, tissue, organ, biological fluid, or organism) at a specific point in time. Mass spectrometry is the [experimental] technique most often used for proteomic analysis." (*Nature.com* definitions)

ProtVec Protein-vectors (or shortly ProtVec) are numerical representations of *protein sequences* that are automatically learned. For more details please see §**2.2** and **2.3**.

PSSM Position-Specific Scoring Matrix or shortly PSSM refers to vector size 20 representing the log-likelihood of the substitution of the 20 types of amino acids at that position in the sequence in the *sequencealignment*.

Random forests Random Forest (RF) is an instance of *machinelearning classifier* ensembling decision trees. In a *classification* problem, the outcome of many simple decision trees (the number of trees is one of the parameters of this model) are aggregated to choose the the most popular *class*.

Recall In a *machine learning classification* task, recall for a *class* is defined as the number of true positives divided by the total number of elements that actually belong to the class. *Micro* and *macro* averaged recalls can be reported (see *micrometrics* and *macrometrics*).

Recurrent Neural Network Recurrent Neural Network (RNNs) are a type of *neural networks* designed for modeling and prediction over sequential data. RNNs take the temporal relation into account by adding feedback connections allowing for conditioning on all previous time steps (for more details please see §**1.2**).

Reference genome Reference sequence refers to a representative set genes of a species

Representation (data) Representation refers to a form that the data on the machine is stored and processed.

Resamples Getting samples from data with replacement.

Rheumatoid arthritis "Rheumatoid arthritis is an autoimmune disease that is characterized by inflammation of the joints and the subsequent destruction of cartilage and erosion of the bone. Patients with rheumatoid arthritis are treated with drugs that suppress the immune system." (*Nature.com* definitions)

RNA "Ribonucleic acid (RNA) is a polymer of 4 nitrogenous bases of guanine (G), uracil (U), adenine (A), and cytosine (C) with essential roles in coding, decoding, regulation and expression of *genes*.

RNN See *Recurrent Neural Network*

Secondary structure of proteins Protein secondary structure is a 3D pattern of the local segments of proteins.

Semantic Semantics refers to the study of meaning in languages.

Semi-supervised learning Semi-supervised learning is the category of machine learning approaches considered as a transition between *unsupervised learning* and *supervised learning*. In semi-supervised learning, we benefit from both input-output example pairs and examples without any assigned output, where usually the second set is much larger.

Sentiment Sentiment refers to the attitude or opinion toward something. Automatic sentiment analysis of text data is one of the crucial topics in language processing.

Sequence A chain of elements.

Sequencing (genetics) Sequencing refers to the process of determining the DNA sequence.

Sub-sequence Sub-sequence is a sequence, which is part of a longer sequence.

Subword Subword refers to parts of a word, which can be meaningful (a lemma) or without meaning.

Supervised learning In machine learning, supervised learning refers to mapping the input to the output (e.g., a category) based on input-output pairs of examples.

Syntax syntax refers to a set of rules governing the structure of sentences in a language.

Synthetic dataset Synthetic data refers to a type of *datasets* that is artificially generated, usually for the *evaluation* purpose.

Taxonomy Taxonomy refers to the hierarchical grouping of organisms into categories based on different properties, including size, shape, and gene sequences.' (*Nature.com* definitions)

Taxon A group of one or more organisms of any rank in *taxonomy*.

Tokenization Tokenization refers to the process of segmenting a sentence into tokens

Token A string of characters between spaces or punctuations (≈ word).

Unsupervised learning Unupervised learning is the category of machine learning approaches drawing inference purely from the input data without having access to the metadata.

Number Sets

$\mathbb{R}$	Real numbers
Ω_l	Embedding space of language l

Other Symbols

$[\bullet \bullet \ldots \bullet]$	In figures, represents a vector representation in the neural network
$\circ$	The element-wise product
$\langle \mathbf{a} \, , \mathbf{b} \rangle$	The inner product of vector **a** and **b**
$\otimes$	In figures, dropout visualization in the neural network
$\vec{x}$	Vector x
$cos(\vec{x}, \vec{y})$	The cosine similarity between vectors $\vec{x}$, $\vec{y}$
$D_{KL}(P \parallel Q)$	KL divergence between distributions P and Q
$m \pm \sigma$	Mean ± standard deviation

Acknowledgments

I would like to express my sincere thankfulness to my research advisor and dissertation chair, Professor Mohammad R.K. Mofrad, for guiding and supporting me much throughout this doctoral research. He has set an example of excellence as a mentor, researcher, instructor, and role model. It has been a great honor for me to be part of his group at Mofrad Lab, where I learned many new and interesting things. I have to thank him for being very much flexible with me in many aspects. Besides my advisor, I would like to thank Professor Shaofan Li, Professor David Bamman, and Professor Joan Bruna Estrach, my qualification exam and dissertation committee members at UC Berkeley for their great and continuous support and encouragement.

Thanks to the unique opportunity I had been given by my advisor Professor Mofrad, I had the chance to enrich my dissertation in close collaborations with the pioneers in the areas of metagenomics, as well as, natural language processing. I would like to thank Professor Alice C. McHardy, the head of the Department of Computational Biology at the Helmholtz Center for Infection Research, who co-supervised me in several research projects I carried out during my PhD. I thank her very much for her great supports, encouragements, and teachings. I was very fortunate to work with her. I have been also pleased to closely work with Professor Hinrich Schütze (the head of the center for information and language processing at the University of Munich) who co-supervised my PhD projects related to natural language processing. I deeply thank him for his supports and teachings, in many regards.

Furthermore, I thank the graduate student advisors of the AS&T program, Ariana Castro and Catalina Estrada, Berkeley International Office, and UC Berkeley Graduate Division for their exceptional care for the students.

In addition, I would like to thank my colleagues at Mofrad lab who contributed to my PhD work with their advice and supports, **Kiavash Garakani**, **Mohammad Soheilypour**, **Mohadesseh Peyro**, **Ruhollah Mousavi Baygi**, **Zeinab Jahed**, **Hengameh Shams**, Zainab Haydari, Mehrdad Mehrbod, and Ali Madani also the UC Berkeley students I had the opportunity to collaborate with: Andrew Dickson, Soroush Nasiriany, Kaveh Karbasi, Vikrum Nibber, Amine Mezzour, Sam Njie, and Anderson Lee. Furthermore, I thank my other mentors, friends, colleagues, and the coauthors who supported me throughout this journey from the PhD application preparation to the PhD completion in different forms, including but nor limted to: **Professor Keivan Majidzadeh**, **Professor Mark Finlayson**, Professor Martin Vetterli, Professor Patrick H. Winston (RIP), Professor Marianne Winslett, Professor Jean-Cedric Chappelier, **Ali Vakilian**, Sepideh Mahabadi, **Hessameddin Akhlaghpour**, **Mohsen Mahdavi-Mazdeh**, **Ardavan Saeedi**, **Saeed Mohajeri**, Pedram Pad, Amin Sadeghi, Sam Sinai, Professor Marzyeh Ghassemi, Hassan Pezeshki Modarres, Salman Faraji, **Payam Delgosha**, **Nima Ahmadipour Anari**, **Yadollah Yaghoobzadeh**, Alireza Ghasemi, Amir Shaikhha, Fazel Parsapour, Sam Nariman, **Professor Amir Kermani**, Reza Abbasi-asl, Ali Sanaei, Ali Basiri, Soroush Sarabi, Amin Aghaei, Mohammad Sahrayian, Samaneh Azadi, Professor Mojtaba Azadi, **Mohammad Keshavarzi**, Professor M.R. Alam, Naeem Esfahani, **Amirhossein Hashemi Astaneh**, Professor Meysam Chamanzar, Professor David Searl, **Meisam Ahmadi**, Hamed Fathi, Elyas Heidari, **Zain Zaidi**, Professor Peter Bartlett, Professor Marti Hearst, Professor Ian Holmes, Professor Sandrine Dudoit, Professor N. Veldhuis, Eduardo Escobar, Professor Ali Jannesari, Arian Hosseini, Behrang Mohit, Benjamin Roth, Heike Adel, Thomas Schäfer, Alena Moiseeva, Professor Peter Rose, Peggy Hobmaier, Gary Robertson, Andreas Bremges, Philipp Münch, Tzu-Hao Kuo, Aaron Weimann, Till-Robin Lesker, Susanne Reimering, Professor Susanne Haeussler, Nina Poerner, **Christoph Ringlestetter**, Fabienne Braune, Jan-Frederik Kassel, Kaya Kim, Omid Rohanian, **Ali Aghebat-Rafat**, Mohammad Samavat, Ali Mollabashi, Professor Ahmad Rezaei, Motreza Karamooz, S. Hassan Zolanvar, and Professor S. Ehsan Seyedabrishami .

I gratefully acknowledge my departmental funding sources at the University of California Berkeley in the first two years, funding resources at the University of Munich (LMU) and Helmholtz Center for Infection Research. I also thank Google for the travel award at the ICML 2016, sponsoring the best poster as well as the best paper awards at the Computational Biology and Multilingual methods workshops, respectively at ICML 2016, and NAACL-HLT 2016.

I sincerely thank my dearest parents for their unconditional love, support, and the efforts I could never compensate. You have been always very kind to me and I owe you my life. I thank my dear parents-in-law, our grandparents, Azizjoon and Mamanjoon, and my extended siblings: Mahdi, Hamed, Jalal, Elaheh, Meisam, Meghdad, Ms. Mahboobeh, and Ms. Mozhdeh, and their lovely sons and daughter: Arsha, Hossein, Mahdi, Ali, Fatemeh,and Sajjad, for their always great encouragements and forgiving us for being physically far from them.

My special thanks goes to my beloved wife, Meshkat, for her continuous support, encouragements, and the sacrifices she made along the way. Thank you so much, Meshkat! Also, thank you for the great helps in design-related aspects of this work. With you, my life became much lighter and much more colorful!

Chapter 1

Introduction to language processing and deep learning

1.1 Language definition and the "language of life"

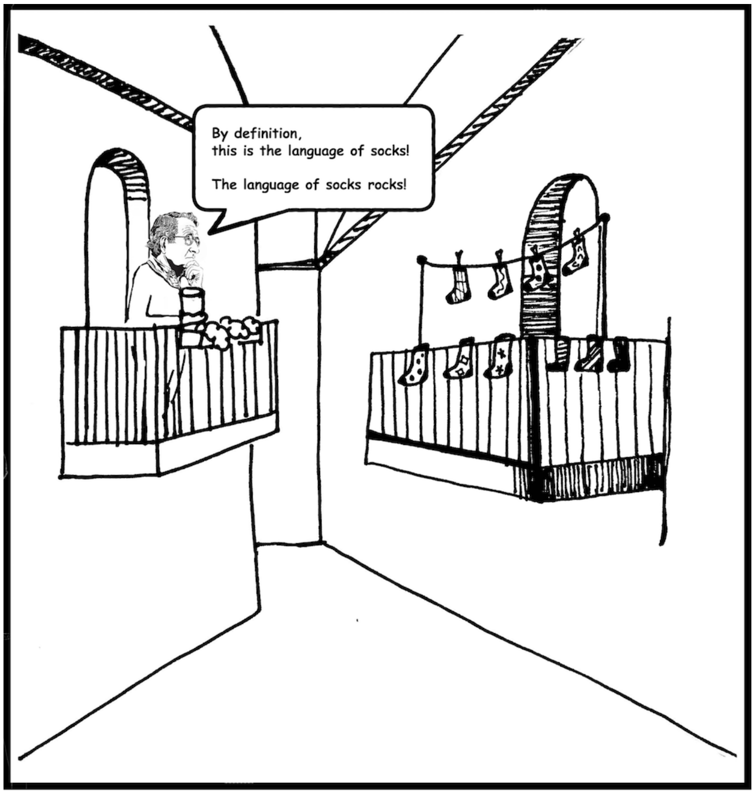
Noam Chomsky introduced a broad definition for language, describing it as "a (finite or infinite) set of sentences, each finite in length and constructed out of a finite set of elements" (Chomsky 2002). Obviously, this definition is not limited to natural languages. An abstract representation of the internal dispositions of the macromolecules of life (i.e., nucleic acids (DNA and RNA) and proteins) satisfies this definition as well, as all of these macromolecules are polymers constructed from a finite set of smaller molecules (Cooper et al. 2000). DNA and RNA are polymers made up of sequences of nucleotides of four distinct types with the alphabetic representations $\{A, T, C, G\}$ and $\{A, U, C, G\}$, respectively, and proteins are polymers made up of sequences of amino acids of 20 different types represented by alphabet characters $\{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$. DNA and RNA are informational molecules, as they carry the genetic instructions needed to make proteins. In contrast to informational molecules, proteins are operational macromolecules, as they contribute to the molecular machinery that carries out the functions that are essential for life. The central dogma of molecular biology describes the relationships among DNA, RNA, and proteins along the flow of genetic information. It states that information flows from the DNA into the RNA in a process called 'transcription', and then further to proteins through a process called 'translation' (Cooper et al. 2000). Even in the terminology, the

[¶] The contents of this chapter have partially appeared in the following publications:

1. Asgari, E., & Mofrad, M. R. (2019). Deep Genomics and Proteomics: Language Model-Based Embedding of Biological Sequences and Their Applications in Bioinformatics. In *Leveraging Biomedical and Healthcare Data* (pp. 167-181). Academic Press.

2. Adel, H. and Asgari, E. and Schütze H. (2017). Overview of Character-Based Models for Natural Language Processing. *Lecture Notes in Computer Science* book series (LNCS, volume 10761), 3–16.

presence of elements such as 'transcription' and 'translation' reflects the notion of viewing this information representation system as the 'language' of life.



Linguists and computational linguists consider a sentence as the output of a complex generative process controlled by certain rules (Jackendoff 1972). They distinguish between syntactic and semantic rules. Generally, syntactic rules govern how the elements are put together to generate well-formed sentences, whereas semantic rules determine the meaning of the resulting sentence. Analogous to what linguists and computational linguists believe about the sequence of words in a sentence, biologists believe that protein and nucleotide (DNA and RNA) sequences are not merely one-dimensional strings of symbols. These sequences encode a lot of information about molecular structure and functions in themselves (Cooper et al. 2000). The structures and functions of macromolecules are interesting for us as they can provide information about genotypes, phenotypes, diseases and even treatments for diseases. Similar to the complex syntax and semantics of natural languages, certain biophysical and biochemical "grammars" dictate the formation of biological sequences. Thus it would be natural to adopt/develop methods in language processing to gain a deeper understanding of how functions and information are encoded within biological sequences, which is one of the main goals of bioinformatics (Yandell and Majoros 2002; Searls 2002; Asgari and M. R. Mofrad 2015).

Bioinformatics and natural language processing

Since this research is at the intersection of bioinformatics and natural language processing (NLP), we begin by defining these. Bioinformatics is defined as *"conceptualizing biology in terms of macromolecules (in the sense of physical-chemistry) and then applying informatics techniques (derived from disciplines such as applied maths, computer science, and statistics) to understand and organize the information associated with these molecules on a large*

[**] Noam Chomsky's cartoon face is taken from SolissClothing's design.

scale" (Luscombe et al. 2001). Some of the primary data types that bioinformaticians deal with are genome sequences, macromolecular structures, the results of functional genomics, scientific literature texts, taxonomy information and interaction networks of macromolecules (Luscombe et al. 2001). Natural language processing can be defined as conceptualizing human languages in terms of written texts. Similar to bioinformatics, NLP researchers use computer science and statistical approaches to understand and manipulate texts by using machines (Chowdhury 2003), which can be very challenging because of the ambiguities that exist in natural languages (Christopher D Manning and Schuetze 1999). Bioinformatics and NLP are research areas that have greatly benefited from each other since their beginnings and there have been always methodological exchanges between them. Levenshtein distance (Levenshtein 1966) and Smith–Waterman (Waterman et al. 1976) algorithms for calculation of string or sequence distances, the use of formal languages for expressing biological sequences (Searls 1993; Searls 2002), training language model-based embeddings for biological sequences (Asgari and M. R. Mofrad 2015) and using state-of-the-art neural named entity recognition architecture (Lample et al. 2016) for secondary structure prediction (Johansen et al. 2017) are some instances of such mutual influence.

1.2 Introduction to deep learning for language processing

The Role Representation in Language Processing

Can a computer automatically comprehend a piece of English text to find documents with similar content or automatically translate the given document to French? These types of tasks constitute the area with which Natural Language Processing (NLP) is mainly concerned. The purpose of NLP is to design algorithms allowing computers to understand natural languages for performing specific tasks (e.g., information retrieval, machine translation, and sentiment analysis). When we want to discuss a complex concept with an audience unfamiliar with the topic, we model or represent the concept within a framework that is understandable for the audience. The same logic applies in presenting a natural language text to a machine. Computers are experts in dealing with numerical values, vectors, and matrices. Thus, the first step in NLP is to vectorize natural language text for computers. Words are the conventional input units of almost all NLP tasks. Therefore, to utilize machines for language processing, we need to find proper vector representations of words that are interpretable by machines. We expect such representations to preserve some indications of similarity and dissimilarity between words. For instance, when we search a phrase in a search engine, we expect the machine to consider words 'formula' and 'equation' to be similar and consider them dissimilar to an irrelevant word like 'cuisine'. Thus, we should attribute similar vector representations to the words 'formula' and 'equation', dissimilar to the vector representation of 'cuisine'. As a reminder vector similarity/distance can be calculated using operations in linear algebra (e.g., dot product, Euclidian distance, and cosine similarity). Of course, semantic similarity is not the only consideration we have in NLP tasks. As an example, part-of-speech tagging is one of the routine NLP tasks, where the goal is to label words with their syntactic part-of-speech (e.g., noun, verb, adverb, etc.). Presumably, when we want to perform part-of-speech tagging, we desire a vector representation incorporating syntactic similarities.

The performance of NLP or in general any machine learning task largely depends on the quality of data representation (also known as feature extraction/engineering), to the extent that recently representation learning became an important research area in machine learning (Bengio et al. 2013; Collobert, Weston, et al. 2011). Deep neural network algorithms effectively allowed for the automatic encoding of data into a proper representation and introduce representation learning as a new field in itself in the realm of machine learning (Bengio et al. 2013). Recent works in the area of representation learning have proposed successful representations of data in computer vision, speech recognition, and natural language processing (Y. LeCun et al. 2015; Graves, Mohamed, et al. 2013; Mikolov, Sutskever, K. Chen, et al. 2013). Similar to the role of textual representations in NLP tasks, representation of biological sequences is key to many bioinformatics tasks, which is facilitated by the recent advances in deep learning (Angermueller et al. 2016).

Ono-to-one representation versus distributed representation

The most straightforward approach to represent textual units (characters, words, or sentences) for a machine is to utilize one element of computing (e.g., a cell in a vector) to represent one entity, in a way that we have a one-to-one mapping from the vector representation to the exact entities. One-to-one representation is also known as *local representation.* One-hot vector encoding is an instance of such a one-to-one representation. Given a set of M words (or any other textual unit) we can represent each word w_i using a vector in the size of M, where the vector has zeros everywhere except at the index of word i having the value of 1, hence calling this representation "one-hot vector representation". Although this representation is straightforward to obtain, it has several drawbacks: (i) this representation is not memory-efficient and not scalable when we have a large collection of textual elements, (ii) this representation does not take the similarities into account, where the similarity of vectors is defined based on their dot product or cosine similarity. In such a space of text representation all word representations of distinct words are orthogonal to each other ($\langle \mathbf{w_i}\, \mathbf{w_j} \rangle = 0 \Leftrightarrow i \neq j$ and $\langle \mathbf{w_i}\, \mathbf{w_j} \rangle = 1 \Leftrightarrow i = j$).

An alternative to the one-hot vector representation approach is the use of distributed representation. Distributed representations are "many-to-many", meaning that (i) each textual unit is represented through a weighted combination of many computing elements, (ii) each computing element gets involved in the representation of many textual units (Geoffrey E Hinton 1984). For an instance manually creating such a representation, one may represent the word w_i in a natural language by a vector size 2, where the first element indicates the polarity score of the word (a real value between -1 and 1) and the second element is the concreteness score of a word (a real value between -1 and 1) that are assigned by a team of experts. This way, similar words (in terms of having similar polarity and concreteness scores) would have similar representations. Distributed representations in comparison with the one-hot vector representations are much more desired for computational tasks because they are more memory efficient and can incorporate the similarities of textual units. In traditional machine learning, feature engineering played a crucial role in obtaining better performance. However, the manual designs of features/representations can be challenging, as they require domain knowledge and sufficient human resources, which are not always granted. Additionally, manually designed features are subject to be bias or incompleteness. This motivates automatic feature extraction.

Machine learning tasks can significantly benefit from the automatic learning of distributed representation. Recently, deep learning methods have achieved very high performances in many areas, including NLP, via training end-to-end models that do not require traditional task-specific feature engineerings. Next, I will provide a brief introduction to such models.

Introduction to neural networks models for language processing

Recently, neural networks models achieved state-of-the-art performance in many natural language processing tasks (Collobert, Weston, et al. 2011; Devlin et al. 2018) via end-to-end learning eliminating traditional feature engineerings. Biological neural networks inspired the design of artificial neural networks. Almost all sorts of neural networks, in an abstract view, are multi-layered models consisting of 3 main types of layers: (i) input layer dealing with the problem domain, (ii) intermediate layer(s) mapping the problem domain to an intermediate distributed representation, and (iii) an output layer dealing with the solution domain. The intermediate layers are supposed to apply non-linear transformation(s) on the input data to prepare a proper representation before the final transformation to the target output. Finally, given the intermediate distributed representation, the output layer provides the final outputs of the model in the solution domain.

Order and Granularity: The input can be either viewed as a sequence of textual units or as a bag (a set of items with repetition) of them. Despite the simplicity, the bag-of-words model has been proven to be very successful in many text classification tasks (Arora et al. 2016). Another consideration is the choice of the atomic unit of text in the input, which is not necessarily word and can be byte (Gillick et al. 2016), character (J. Lee et al. 2016), character n-gram (Schütze et al. 2016), byte-pairs (Sennrich et al. 2016), morpheme (Kirchhoff et al. 2006), etc.

There exist a variety of models for the intermediate layers to encode the input data; examples are convolutional layers (C. N. d. Santos and Gatti 2014), recurrent layers (Sutskever, Martens, et al. 2011), capsule network (Sabour et al. 2017; W. Zhao et al. 2018), and various combinations that different models together can make. Here, I describe the main basic models used within the framework of this dissertation.

Multi-layer preceptron neural network

Multi-Layer-Perceptrons (MLP) also known as fully-connected feedforward neural network is the basic neural architecture approximating function f^*, mapping the input vector $\boldsymbol{x}$ to the output $\boldsymbol{y}$ through non-linear transformations parameterized with θ:

$$\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \theta)$$

The function f is called a network as this is the result of connecting multiple non-linear layers f_i's:

$$f(\boldsymbol{x}; \theta) = f_M(\dots f_3(f_2(f_1(\boldsymbol{x})))),$$

where M is the number of non-linear transformations, and each f_i can be parameterized as follows:

$$f_i(\boldsymbol{x_i}; \theta_i) = g_i({\boldsymbol{x_i}}^T W_i + b_i),$$
$$\theta_i = \{W_i, b_i\}$$

where $\boldsymbol{x_i} \in \mathbb{R}^{D_i}$ is the input to the i_{th} layer and $W \in \mathbb{R}^{D_i \times D_{i+1}}$ is a linear transformation from the input dimension D_i to the dimension of the output of the current layer D_{i+1} and $b_i \in \mathbb{R}^{D_{i+1}}$ is the bias term in the affine transformation. The g_i is a non-linear activation function of the i^{th} layer.

Input and output of MLP: The MLP neural network architecture is depicted in Figure 1.1. The other name of this network is feedforward network, as the information flow is forwarded from the input $\boldsymbol{x}$ to the output $\boldsymbol{y}$ through f_i's. The MLP architecture is mainly designed for bag-of-word input features, as the contextual and global structure of a sequence cannot be easily modeled in this architecture. However, the extended versions of MLP, i.e., convolutional and recurrent neural networks can take the sequential and global structure into account. The MLP can be used to regress real values (regression) or to predict a category (classification); the non-linear activation function of the last layer can be determined according to this objective. The g_i's are commonly chosen from the following differentiable non-linear functions:

- **Sigmoid (σ)**: sigmoid function maps any given $x \in \mathbb{R}$ to a value between 0 and 1:
$$g_i = \sigma(\boldsymbol{x}) = \frac{1}{1+e^{-x}},$$
which is usually favored as the ultimate transformation in a binary classification setting. Considering a threshold of 0.5 on the out put of sigmoid it can be binarized for prediction between two classes.

- **Tanh**: Tangent hyperbolic function is also a non-linear function that commonly used for transfering the input $x \in \mathbb{R}$ to a value between -1 and 1 and can be formulated as follows:
$$g_i = tanh(\boldsymbol{x}) = \frac{2}{1+e^{-2x}} - 1 = 2\sigma(2\boldsymbol{x}) - 1,$$

- **Rectified linear unit (ReLU)**: ReLU is a very simple, but in practice a very effective non-linear transform used in neural networks. ReLU for any given input x gives an output x if x is positive and 0 otherwise:
$$g_i = ReLU(\boldsymbol{x}) = max(x, 0).$$

- **Softmax**: In the case of multi-class classification, the softmax activation function is used at the last layer to produce the probability vector that can be regarded as representing posterior probabilities (Goodfellow et al. 2016).:
$$g_i = softmax(x_k) = \frac{exp(x_k)}{\Sigma_{x_l \in \boldsymbol{x}} exp(x_l)}.$$

Learning: Back-propagation algorithm and stochastic gradient decent

The forward propagation of information from input $\boldsymbol{x}$ to the predicted output $\boldsymbol{y}$ has been discussed so far. Subsequently at the output layer we can measure the cost function $J_\theta(\boldsymbol{y}, \hat{\boldsymbol{y}})$ based on the gold output $\boldsymbol{y}$ and the predicted output $\hat{\boldsymbol{y}}$. The back-propagation algorithm (Rumelhart et al. 1986) allows the backward propagation of information from the cost

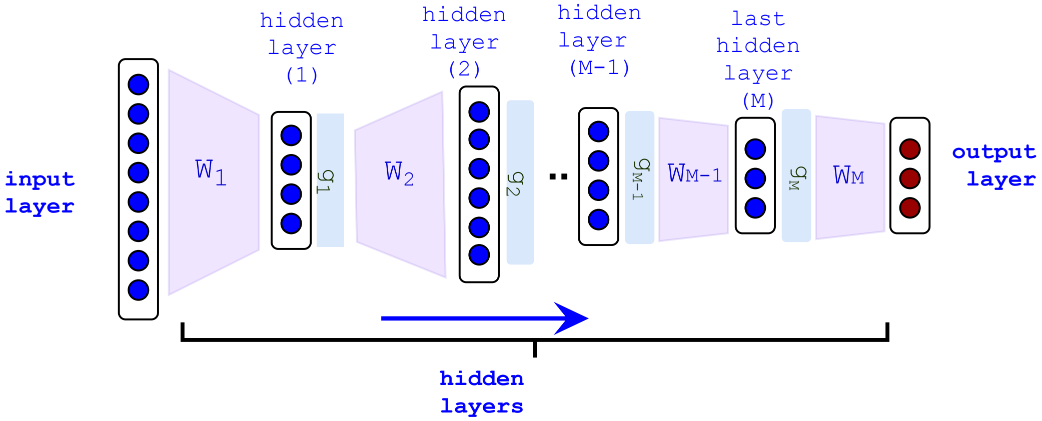


Figure 1.1. Architecture of Multi-Layer-Perceptrons (MLP) Neural Network with M hidden layers and non-linear activation functions of g_i's on top of each linear transformation using W_i matrices.

function J_θ calculated at the output back to the first hidden layer in the network through the gradients over $\theta_{1:M}$. Then Stochastic Gradient Descent (SGD) algorithm (Y. A. LeCun et al. 2012) is utilized to perform parameter optimization minimize the cost function using the calculated gradient in the back-propagation. This approach of learning (the combination of back-propagation and SGD) is not limited to the MLP, but applicable to any function whose derivative is defined. Different variations of SGD are proposed in the literature, including Momentum (Polyak 1964) and Adam (Kingma and Ba 2015), allowing for faster learning and more optimal parameters. To avoid overfitting, usually *earlystopping* and also *dropout* at hidden layers (N. Srivastava et al. 2014b) are performed. Early stopping refers to stop the process of parameter optimization when further optimization only improves the performance over the training and not the test data. Dropout is a simple but powerful regularization approach for neural networks by randomly dropping out hidden units.

The loss function will be also determined based on the problem-setting. In the case of classification, the cross-entropy is the most common loss function. Given that the gold label $\boldsymbol{y}$ presented using one-hot representation and the output layer is defined as softmax generating $\hat{y}$, the cost function J_θ between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ can be defined as follows:

$$J_\theta(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \Sigma_{i=1}^{|\boldsymbol{y}|} - (y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_j))$$

Convolutional Neural Networks (CNN)

The development of Convolutional Neural Networks has been mainly inspired by the function of human visual system containing several filters and detectors of different patterns (Fukushima 1980). Convolution operation between two given functions p and q can be defined as follows:

$$(p * q)[i] = \Sigma_{j=-M}^{M} p[i - j].q[j],$$

where $p * q$ can be regarded as a filtered version of p by function q. Convolutional layers in the neural networks are trainable filters applied in the spatial or temporal axis of input.

Given an input sequence of textual units (e.g., words) $\boldsymbol{x_i'}s \in \mathbb{R}^d$ within a sentence of length $N, (N \geq i \geq 1)$, a 1D convolution of window size c, can be defined as follows (Y. Kim 2014):

$$h_i = g(W^T \boldsymbol{x}_{i:(i+c-1)} + b),$$

where $\boldsymbol{x_{i:i+c-1}} \in \mathbb{R}^{dc}$ represents the concatenation of $\boldsymbol{x_i}$ vectors, and $W \in \mathbb{R}^{dc}$ is the convolutional filter. By putting the h_i's of a sentence together ($i \leq N - c + 1$), we can generate a new representation of sentence $S = [h_1, h_2, \dots, h_{N-c+1}] \in \mathbb{R}^{N-c+1}$.

Max-pooling: a common approach for the representation of textual data after a convolutional filter is applying $max - over - time$ filter to pick a value in S containing the maximum value. The main idea is to capture the most critical activation during the time.

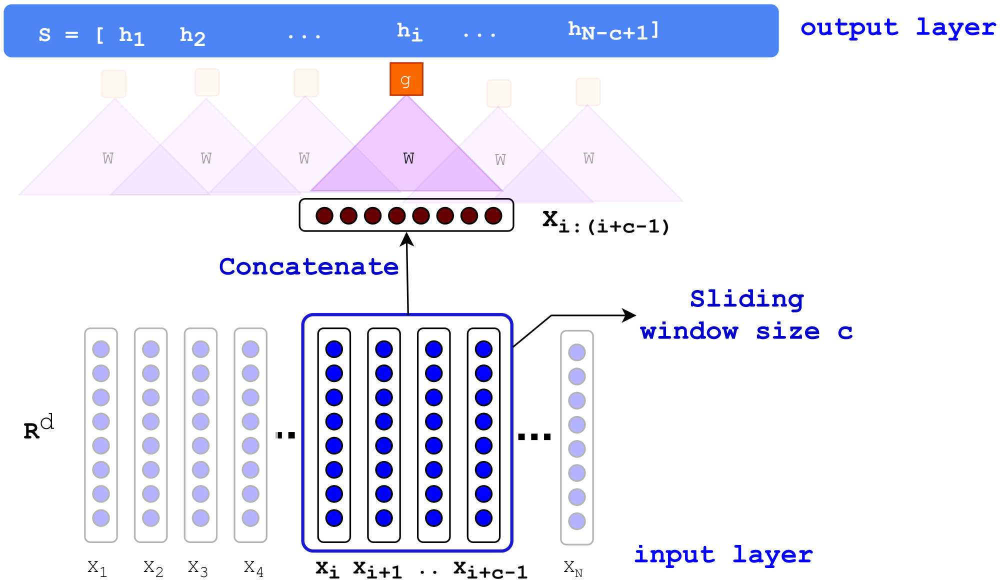


Figure 1.2. The architecture of Convolutional Neural Network (CNN) with a sliding window size of c applied on a sequential data (text). The X_i (the input of CNN at time i) and $c - 1$ of its left neighbors (X_{i+1}, X_{i+1}, …, X_{i+c-1}) are concatenated to form a single vector $X_{i:(i+c-1)}$. Subsequently, $X_{i:(i+c-1)}$ through a non-linear transformation $g(W^T \boldsymbol{x}_{i:(i+c-1)} + b)$ produces h_i (the output of CNN at time i). Next, the window is shifted to produce h_{i+1}.

The architecture of CNN is depicted in Figure 1.2. In practice, multiple filters (multiple W's) are applied to the input sequence to capture multiple views from a sequence. Since functions applied in CNN are also differentiable, the training of CNN layer is the same as learning of the MLP discussed above. The CNN layers allow for the integration of local context in specific window sizes and stacking of CNN layers on top of each-other facilitates moving from a local context to a more global context.

Neural Sequence Modeling

Language modeling Language modeling is the task of assigning a probability $P(w_1, w_2, \ldots, w_N)$ to a given sequence of textual unit (words, phrases, or sentences) $w_1, w_2, \ldots, w_N$. Language modeling is a very important component in many NLP applications, in particular the applications containing language generation or the evaluation of text correctness, e.g., machine translation, auto-completion of text, chat-bot, etc. Using the chain rule, this probability can be written as follows:

$$P(w_1, w_2, \ldots, w_N) = P(w_1) \times P(w_2|w_1) \times P(w_3|w_1, w_2) \times \ldots \times P(w_N|w_1, \ldots, w_{N-1})$$

For the simplicity of modeling in practice k^{th} order Markov model is considered, i.e., the w_t only depends on the k preceding textual units:

$$P(w_t|w_1, \ldots, w_{t-1}) = P(w_t|w_{t-k}, \ldots, w_{t-1}),$$

which can be estimated by counting of n-gram of textual units, e.g., for the $k = 2$:

$$P(w_t|w_1, \ldots, w_{t-1}) = P(w_t|w_{t-1}, w_{t-1}) = \frac{count(w_{t-2}, w_{t-1}, w_t)}{count(w_{t-1}, w_t)},$$

where $count(w_{t-2}, w_{t-1}, w_t)$ is the number of times the sequence of w_{t-2}, w_{t-1}, w_t is observed in the whole collection of texts. In a more broad definition, $w_1, \ldots, w_{t-1}$ is the left context of w_t. In other words, in language modeling, we study the probability of w_t given its context. By increasing the k, the model becomes more powerful in sequence modeling. However, on the other hand, the model becomes exponentially larger, and a larger collection for estimating would be needed. Recurrent neural networks and distributed representation are alternative models for the sequence modeling that can be used for the language modeling task as well.

Recurrent neural network (RNN): Recurrent neural networks extend the MLP architecture for the modeling of sequential data. RNNs take the temporal relation into account by adding feedback connections allowing for conditioning on all previous states. The RNN architecture is depicted in Figure 1.3. Given the sequence input of $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}\} \in \mathbb{R}^{d_x}$ (e.g., sequence of words or subwords), where d_x is the dimension of input text units, a single time step in the RNN can be formulated as follows:

$$\begin{aligned} h_t &= tanh(W_{hh}h_{t-1} + {x_t}^T W_{xh}) \\ \hat{y_t} &= softmax({h_t}^T W_{hy}), \end{aligned} \tag{1.1}$$

where the $h_t \in \mathbb{R}^{d_h}$ is the hidden state of RNN at time step t, $W_{hh} \in \mathbb{R}^{d_h \times d_h}$ determines the contribution of the previous hidden state h_{t-1} in the current hidden state, and $W_{xh} \in \mathbb{R}^{d_x \times d_h}$ determines the contribution of input x_t in the h_t. Subsequently, a transformation of h_t through $W_{hy} \in \mathbb{R}^{d_h \times d_y}$ as each time step produces the output $\hat{y_t}$. The output can be different in different problem settings. As discussed earlier in the case of multi-class classification $softmax$ is used and d_y is the number of classes we have at the output (to be comparable with the one-hot encoding of the labels). Since in the language modeling (prediction of the next word/subword) the output is also limited to the list of vocabulary, the problem setting is the same as multi-class classification and $softmax$ is used in the last layer. Please note that the W_{hh}, W_{xh}, and W_{hy} remain constant across the time steps.

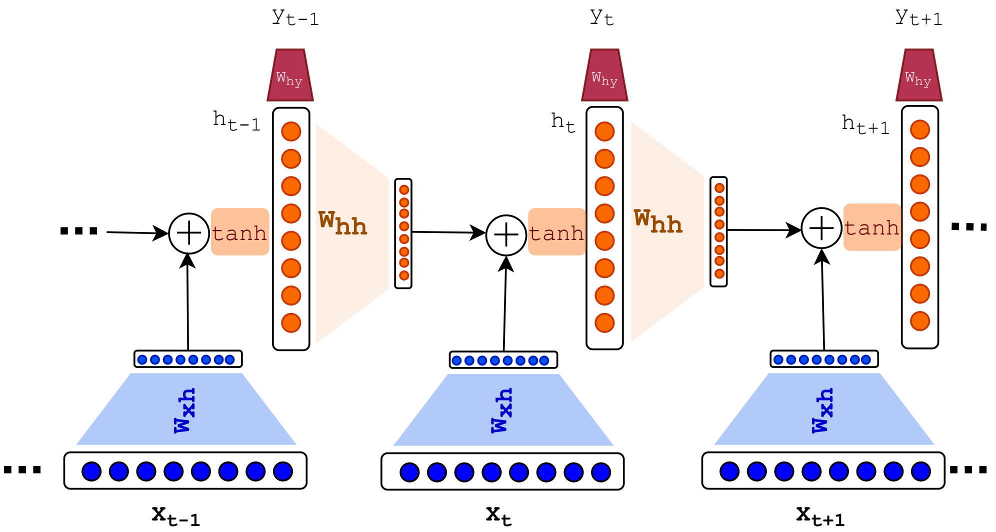


Figure 1.3. The architecture of Recurrent Neural Network (RNN) is shown around the time step t. At each time step t, the X_t (the t^{th} input element of the sequence) maps to a vector in $\mathbb{R}^{d_h}$ through W_{xh}. Next, to produce h_t (the hidden state at time t), ${x_t}^T W_{xh}$ vector is added to a transformed version of h_{t-1} (through W_{hh}) and a non-linear transformation (here $tanh$) is applied on top, i.e., $h_t = tanh(W_{hh}h_{t-1} + {x_t}^T W_{xh})$. h_t at each step produces the output y_t through a transformation W_{hy}. Note that all W's (W_{xh}, W_{hh}, W_{hy}) are linear transformations shared across time steps.

The training process of an RNN is also similar to the MLP network, where we perform the back-propagation through time (Werbos et al. 1990). Although RNNs are supposed to take the full context of the input sequence into account, vanishing gradient (when the gradient is very small, and the weight will not be updated anymore) can occur in their training. Succeeding variants of RNNs, namely Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Unit (Chung, Gulcehre, et al. 2015) resolved this issue and allowed for efficient training of RNNs to incorporate long term dependencies.

Language model-based representation learning

Transfer learning in machine learning refers to the use of the solution in a problem setting (source problem) to solve a different problem (target problem). Using a neural network trained for a specific task for another task is also an instance of transfer learning. The

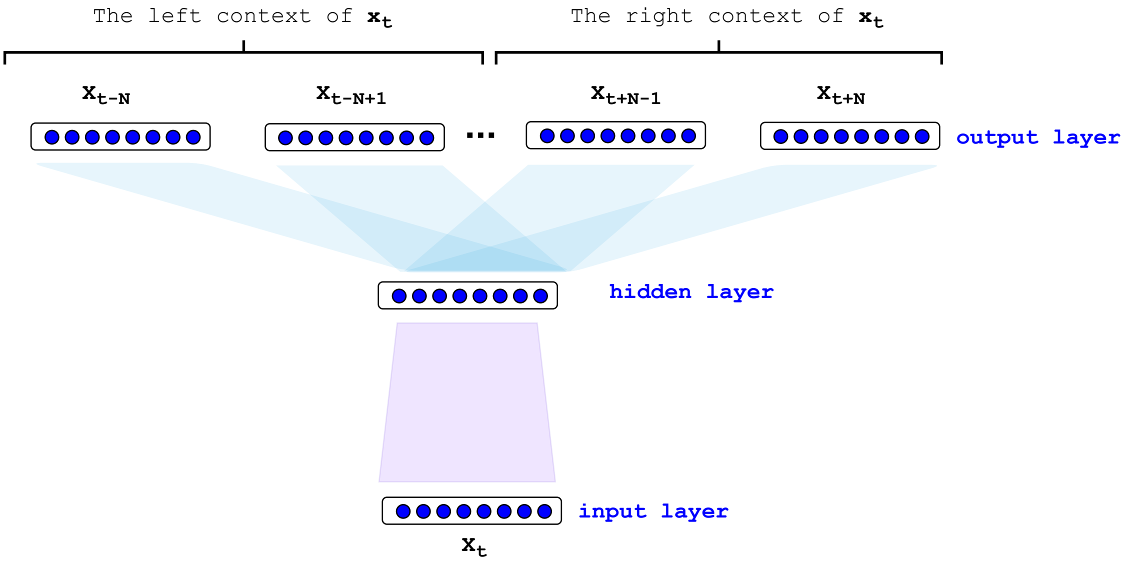


Figure 1.4. Skip-gram neural network for the training of language model-based embedding. In this network for a given word the context words are predicted.

more general the source problem, the more target problems can benefit from the pre-trained network. Thus, an ideal scenario for a source problem would be a case with (i) enough training instances and (ii) general purpose task; then we train a proper network on the source problem and fine tune it for the target problem using a less number of training instances. This way, the pre-trained network serves as a prior knowledge from the source problem to the target problem.

Language modeling (prediction of the next word based on the given context) is an ideal task for the purpose of transfer learning because: (i) the training of a neural language model only requires the raw text without any meta-data or annotation (ii) language modeling is a very general task containing information about syntax and semantics of a language. Recently, transfer learning from the language modeling domain became very popular in NLP and successfully obtained state-of-the-art performance in many tasks. Here we describe the Skip-gram (Mikolov, Sutskever, K. Chen, et al. 2013), one of the most successful architecture to obtain the text representations, which is also used within the framework of this dissertation.

The objective of Skip-gram neural network (depicted in Figure 1.4) is analogous to the objective of the language modeling task. The difference is that the input and output are interchanged, the Skip-gram predicts the surroundings (context) for a given textual unit. Formally, the objective of skip-gram is to maximize the following log-likelihood:

$$\sum_{t=1}^{M} \sum_{c \in [t-N, t+N]} \log p(w_c \mid w_t), \tag{1.2}$$

where N is the surrounding window size around word w_t, c is the context indices around index t, and M is the corpus size in terms of the number of available words and context pairs. We parameterize this probability of observing a context word w_c given w_t by using word embedding:

$$p(w_c \mid w_t; \theta) = \frac{e^{v_c \cdot v_t}}{\sum_{c' \in \mathcal{C}} e^{v_{c'} \cdot v_t}}, \tag{1.3}$$

where $\mathcal{C}$ denotes all existing contexts in the training data. However, iterating over all existing contexts is computationally expensive. This issue can be efficiently addressed by using negative sampling. In a negative sampling framework, we can rewrite Equation 1.2 as follows:

$$\sum_{t=1}^{T} \left[\sum_{c \in [t-N, t+N]} \log \left(1 + e^{-s(w_t,\ w_c)}\right) + \sum_{w_r \in \mathcal{N}_{t,c}} \log \left(1 + e^{s(w_t,\ w_r)}\right) \right], \tag{1.4}$$

where $\mathcal{N}_{t,c}$ denotes a set of randomly selected negative examples sampled from the vocabulary collection as non-contexts of w_t and $s(w_t,\ w_c) = {v_t}^{\top} \cdot v_c$ (parameterization with the word vector v_t and the context vector v_c). The sub-word level Skip-gram, known as fasttext (Bojanowski et al. 2017) improves the word representations by taking character n-grams of the sub-words into consideration in calculating the embedding of a given word. In the fasttext model, the scoring function will be based on the vector representation of n-grams (e.g., $2 \leq n \leq 6$) that exist in textual units (e.g, word), $s(w_t, w_c) = \sum_{x \in \mathcal{S}_{w_t}} v_x^{\top} v_c$.

Motivation for sub-word levels language processing in this dissertation

Natural language processing relies on a preprocessing pipeline, such as the one described in (Gillick et al. 2016) and depicted in Figure 1.5. First, the document is tokenized. This step needs language-specific tokenization tools. The token sequence is then segmented into sentences. Afterward, the syntactic and semantic analysis is performed (usually sentence-wise). Syntactic analysis outputs part-of-speech tags, syntactic dependencies, etc. Semantic analysis extracts named entity tags, semantic roles, etc. The actual natural language processing/understanding (NLP/NLU) task (e.g., answering questions or extracting information) uses features from those preprocessing steps. Tokenization is a critical step in NLP that can have a great impact on the performance of the entire pipeline. Although tokenizable words in many languages seem to be an obvious choice of text processing unit, subword-level and character-level language processing has recently become popular among the NLP community to address the propagation of tokenization errors in the NLP pipeline. Since biological sequences often do not contain even naive segmentation boundaries of functional units, this problem is even more severe in bioinformatics. Thus, subword- and character-level language processing methods can be beneficial for language-agnostic processing of both natural languages and biological sequences. In section §1.3, we provide a brief survey of these new approaches organized by different task types in NLP (Adel et al. 2017).

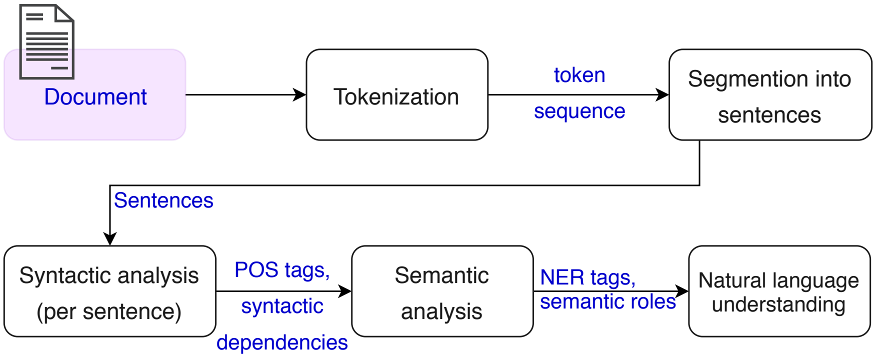


Figure 1.5. Natural language preprocessing pipeline. POS, part of speech. First, the document is tokenized. The token sequence is then segmented into sentences. Afterward, the syntactic and semantic analysis is performed (usually sentence-wise). Syntactic analysis outputs POS tags, syntactic dependencies, etc. Semantic analysis extracts named entity tags, semantic roles, etc. The actual natural language understanding (NLU) task (e.g., answering questions or extracting information) uses features from those preprocessing steps.

1.3 Overview of character-level methods in natural language processing

In the NLP pipeline (shown in Figure 1.5), since every preprocessing step can have deficiencies, the entire pipeline of modules is prone to subsequent errors. Usually, it is hard, inefficient or even impossible to recover from those errors, especially when they occur during tokenization (i.e., in the first step of the pipeline). Although tokenization is easy in many cases in English,[*] it can be very hard for other languages; for example, it is hard for Chinese because the tokens are not separated by spaces, it is hard for German because of compounds and it is also hard for agglutinative languages like Turkish. Therefore, character-based models have the potential of being more robust for NLP. Furthermore, they support end-to-end approaches for text that do not require manual definitions of features, similar to pixel-based models in vision or acoustic signal-based approaches in speech recognition.

In the following, we will present an overview of work on character-based models for a variety of tasks from different NLP areas.[†]

Subword- and character-level models for NLP

The history of character-based research in NLP is long and spans a broad array of tasks. Here, we make an attempt to categorize the literature on character-level work into three classes based on the way they incorporate character-level information into their computational models. The three classes we can identify are: **tokenization-based models**, **bag-of-n-gram models** (known as k-mers in bioinformatics domain) and **end-to-end models** (Schütze et al. 2016). However, mixtures are also possible, such as tokenization-based bag-of-n-gram models or bag-of-n-gram models trained end-to-end.

On top of categorization based on the underlying representation model, we sub-categorize the work within each group into six abstract types of NLP tasks (if possible) to be able to compare them more directly. Abstract definitions of the tasks also help in finding similar tasks in bioinformatics. These task types depicted in Figure 1.6 are as follows:

1. **Representation learning for character sequences:** Work in this category attempts to learn a generic representation for sequences of characters in an unsupervised fashion on large corpora. Learning such representations has been shown to be useful for solving downstream NLP tasks (Collobert, Weston, et al. 2011; Kocmi and Ondrej Bojar 2016; Mikolov, Sutskever, K. Chen, et al. 2013).

2. **Sequence-to-sequence generation:** This category includes a variety of NLP tasks mapping variable-length input sequences to variable-length output sequences. Tasks

[*] There are also difficult cases in English, such as “Yahoo!” or “San Francisco–Los Angeles flights”.

[†] In our view, morpheme-based models are not true instances of character-level models because linguistically motivated morphological segmentation is an equivalent step to tokenization, but on a different (i.e. subword) level. We therefore do not cover much work on morphological segmentation here.

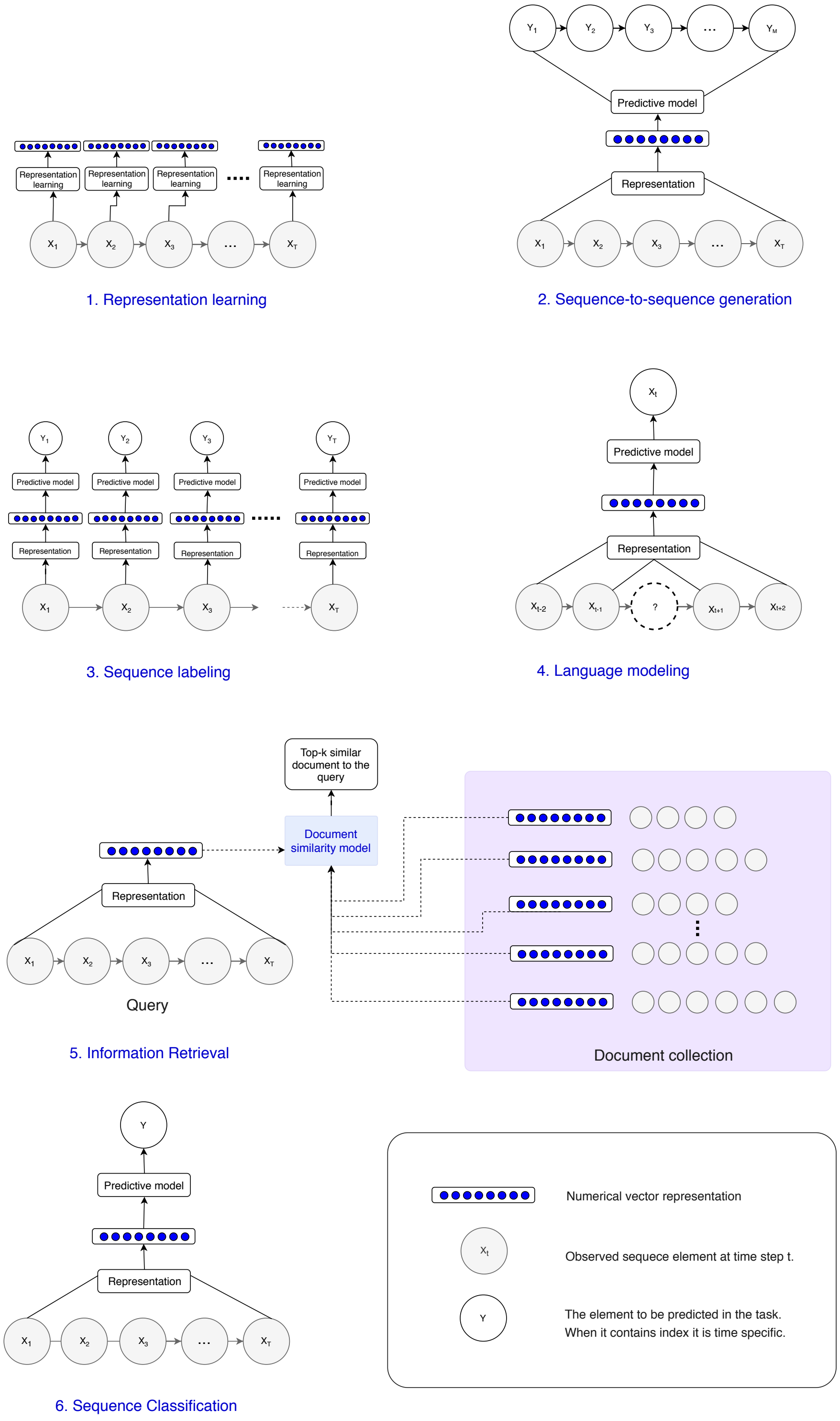

Representation learning
X_1
X_2
X_3
...
X_T
1. Representation learning
Y_1
Y_2
Y_3
...
Y_M
Predictive model
Representation
2. Sequence-to-sequence generation
Predictive model
Representation
Y_T
3. Sequence labeling
X_t
X_{t-2}
X_{t-1}
?
X_{t+1}
X_{t+2}
4. Language modeling
Top-k similar document to the query
Document similarity model
Query
5. Information Retrieval
Document collection
Y
6. Sequence Classification
Numerical vector representation
Observed sequece element at time step t.
The element to be predicted in the task.
When it contains index it is time specific.

Figure 1.6. Abstract visualization of natural language processing (NLP) task types: (1) Representation learning, (2) Sequence-to-sequence generation, (3) Sequence labeling, (4) Language modeling, (5) Information retrieval, and (6) Sequence classification.

in this category include those that are naturally suited for character-based modeling, such as grapheme-to-phoneme conversion (Bisani and Ney 2008; Kaplan and Kay 1994; Sejnowski and Rosenberg 1987), transliteration (Haizhou et al. 2004; K. Knight and Graehl 1998; Sajjad 2012), spelling normalization for historical text (Pettersson et al. 2014), or restoration of diacritics (Mihalcea and Nastase 2002). Machine translation and question answering are other major examples of this category.

3. **Sequence labeling:** NLP tasks that assign a categorical label to a part of a sequence (a character, a sequence of characters or a token) are included within this group. Part-of-speech tagging, named entity recognition, morphological segmentation and word alignment are instances of sequence labeling.

4. **Language modeling:** Another type of task for character-based modeling that has been important for a very long time is language modeling. In 1951, Shannon (Shannon 1951) proposed a guessing game asking "How well can the next letter of a text be predicted when the preceding N letters are known?" This is basically the task of character-based language modeling.

5. **Information retrieval:** The information retrieval task is to retrieve the character sequence that is most relevant to a given character sequence (the query) from a set of existing character sequences.

6. **Sequence classification:** In this type of NLP task, a categorical label will be assigned to a character sequence (e.g., a document). Instances of this type are language identification, sentiment classification, authorship attribution, topic classification and word sense disambiguation.

Tokenization-based approaches

We group the character-level models that are based on tokenization together as a necessary preprocessing step in the category of tokenization-based approaches. Those can be either models with tokenized text as input or models that operate only on individual tokens (such as studies on morphological inflection of words).

In the following paragraphs, we cover a subset of tokenization-based models that are used for representation learning, sequence-to-sequence generation, sequence labeling, language modeling and sequence classification tasks.

Representation learning for character sequences. Creating word representations based on characters has attracted much attention recently. Such representations can model rare words, complex words, out-of-vocabulary words and noisy texts. In comparison to traditional word representation models that learn separate vectors for word types, character-level models are more compact, as they only need vector representations for characters as well as a compositional model.

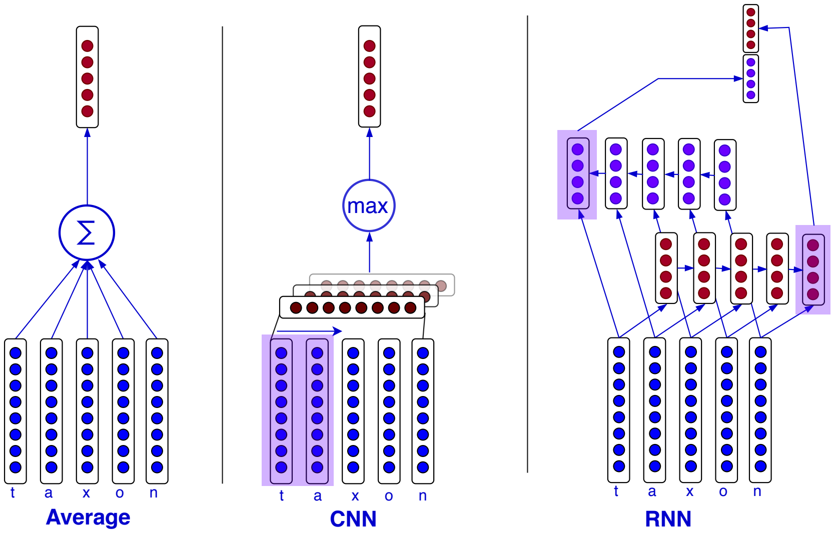


Figure 1.7. Models to calculate word embeddings based on characters

Various neural network architectures have been proposed for learning token representations based on characters. Examples of such architectures are depicted in Figure 1.7 (from left to right): averaging character embeddings, (bidirectional), recurrent neural networks (RNNs) (with or without gates) over character embeddings and convolutional neural networks (CNNs) over character embeddings. Studies on the general task of learning word representations from characters include (X. Chen et al. 2015; Ling, Trancoso, et al. 2015; M.-T. Luong et al. 2013; Vylomova et al. 2016). These character-based word representations are often combined with word embeddings and can be integrated into a hierarchical system, such as hierarchical RNNs (see Figure 1.8) or CNNs (see Figure 1.9) or a combination of both (see Figure 1.10) to solve other task types. We will provide more concrete examples below.

Sequence-to-sequence generation (machine translation). Character-based machine translation is no new topic. Using character-based methods has been a natural way to overcome challenges like rare or out-of-vocabulary words in machine translation. Traditional machine translation models based on characters or character n-grams have been investigated by (Lepage and Denoual 2005; Joerg Tiedemann and Nakov 2013; Vilar et al. 2007). Neural machine translation with character-level and subword units has become popular recently (Costa-Jussa and Fonollosa 2016; M. Luong and Christopher D. Manning 2016; Sennrich et al. 2016; Vylomova et al. 2016). In such neural models, using a joint attention and translation model makes joint learning of alignment and translation possible (Ling, Trancoso, et al. 2015).

Both hierarchical RNNs (Ling, Trancoso, et al. 2015; M. Luong and Christopher D. Manning 2016) (similar to Figure 1.8) and combinations of CNNs and RNNs have been proposed for neural machine translation (Costa-Jussa and Fonollosa 2016; Vylomova et al. 2016) (similar to Figure 1.10).

Sequence labeling. Examples of early efforts on sequence labeling using tokenization-based models include: bilingual character-level alignment extraction (Church 1993), unsupervised multilingual part-of-speech induction based on characters (A. Clark 2003), part-of-speech tagging with subword- or character-level information (Andor et al. 2016; Hardmeier 2016; Ratnaparkhi et al. 1996), morphological segmentation and tagging (Cotterell et al. 2016; T. Mueller et al. 2013) and identification of language inclusion with character-based features (Alex 2005).

Recently, various hierarchical character-level neural networks have been applied to a variety of sequence labeling tasks.

- Recurrent neural networks are used for part-of-speech tagging (depicted in Figure 1.8) (Ling, Dyer, et al. 2015; Plank et al. 2016; Zhilin Yang et al. 2016), named entity recognition (Lample et al. 2016; Zhilin Yang et al. 2016), chunking (Zhilin Yang et al. 2016) and generation of morphological segmentation or inflection (Cao and Rei 2016; Faruqui et al. 2016; Kann, Cotterell, et al. 2016; Kann and Schuetze 2016a; Kann and Schuetze 2016b; Rastogi et al. 2016; Linlin Wang et al. 2016; Yu et al. 2016). Such hierarchical RNNs are also used for dependency parsing (Ballesteros et al. 2015). This work has shown that morphologically rich languages benefit from character-level models in dependency parsing.

- Convolutional neural networks are used for part-of-speech tagging (shown in Figure 1.9) (C. N. d. Santos and Zadrozny 2014) and named entity recognition (C. N. d. Santos and Guimaraes 2015).

- A combination of RNNs and CNNs is used, for instance, for named entity recognition (shown in Figure 1.10) (Chiu and Nichols 2016; V et al. 2016).

Language modeling. Earlier work on subword language modeling has used morpheme-level features for language models (Bilmes and Kirchhoff 2003; Ircing et al. 2001; Kirchhoff et al. 2006; M. Ali Basha Shaik et al. 2013; Vergyri et al. 2004). In addition, hybrid word and n-gram language models for out-of-vocabulary words have been applied in speech recognition (Hirsimaki et al. 2006; Kombrink et al. 2010; Parada et al. 2011; M Ali Basha Shaik et al. 2011). Furthermore, characters and character n-grams have been used as input to restricted Boltzmann machine-based language models for machine translation (Sperr et al. 2013).

More recently, character-level neural language modeling has been proposed extensively (Bojanowski et al. 2017; Botha and Blunsom 2014; Y. Kim et al. 2016; Ling, Dyer, et al. 2015; Mikolov, Sutskever, Deoras, et al. 2012; M. Ali Basha Shaik et al. 2013; Sperr et al. 2013). Although most of this work has used RNNs, some architectures combine CNNs and RNNs (Y. Kim et al. 2016). Though most of these studies combine the output of the character model with word embeddings, the authors of (Y. Kim et al. 2016) report that this does not help them for their character-aware neural language model. They use convolution over character embeddings followed by a highway network (R. K. Srivastava et al. 2015) and feed its output into a long short-term memory network that predicts the next word via a softmax function.

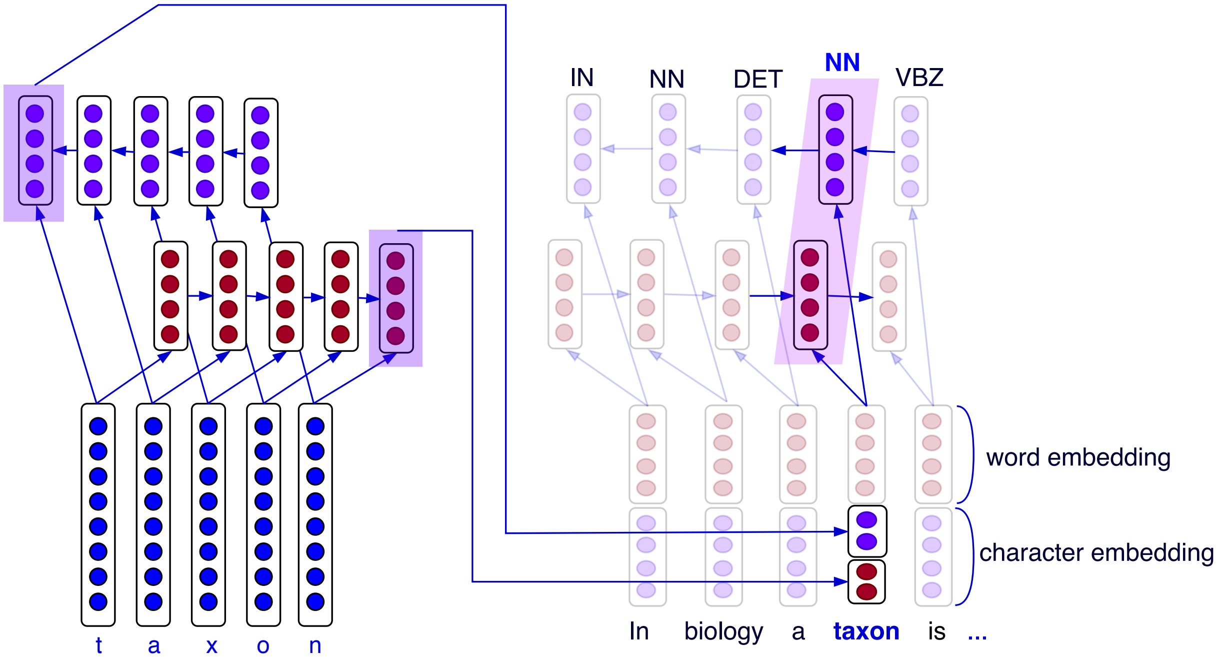


Figure 1.8. Hierarchical RNN for part-of-speech tagging with character embeddings as input

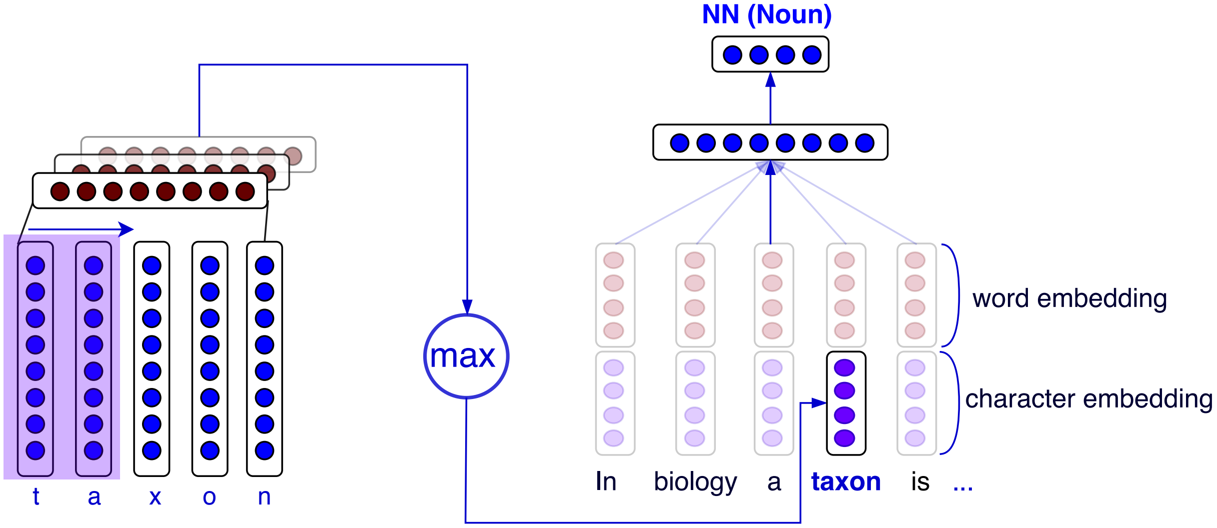


Figure 1.9. Hierarchical CNN + MLP for part-of-speech tagging with character embeddings as input

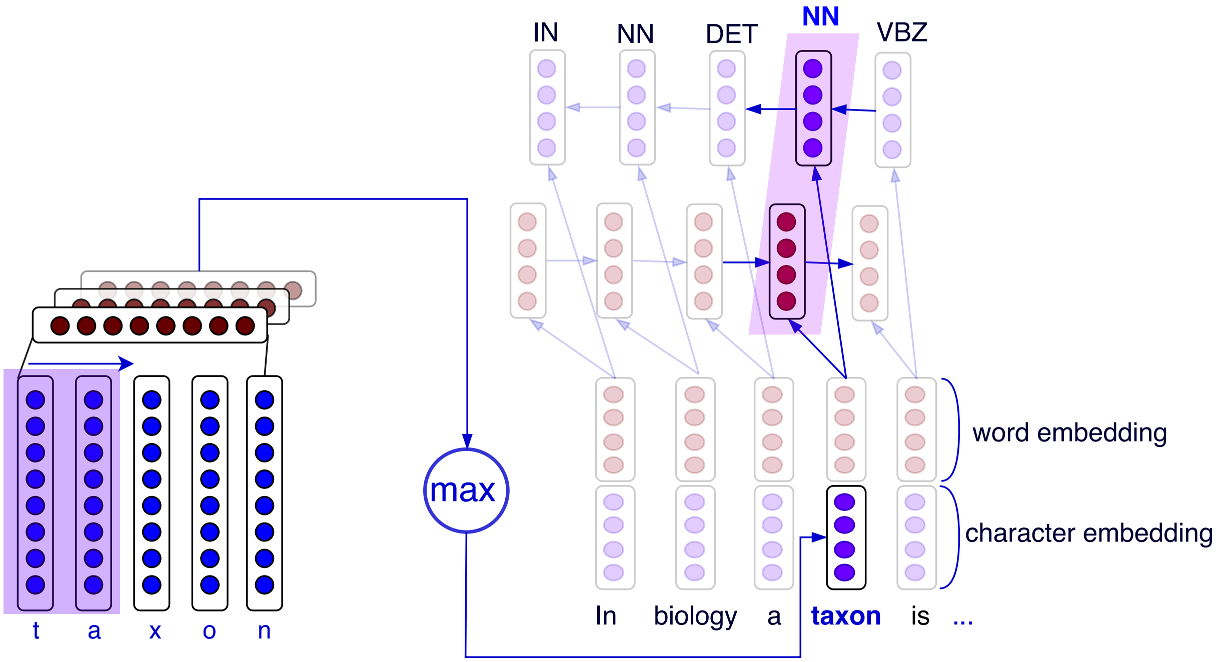


Figure 1.10. Hierarchical CNN + RNN for part-of-speech tagging with character embeddings as input

Sequence classification. Examples of tokenization-based models that perform sequence classification are CNNs used for sentiment classification (C. N. d. Santos and Gatti 2014) and combinations of RNNs and CNNs used for language identification (Jaech et al. 2016).

Bag-of-n-gram Models

Character n-grams have a long history of being known as the features of specific NLP applications, such as information retrieval. However, work has attempted to represent words or larger input units, such as phrases, with character n-gram embeddings. These embeddings can be within-token or cross-token (i.e., no tokenization is necessary).

Although such models learn or use character n-gram embeddings from tokenized text or short text segments, in order to represent a piece of text, the character n-grams that occur are usually summed without the need for tokenization. For example, the phrase "Berlin is located in Germany" is represented by character 4-grams as follows: "Berl erli rlin lin_ in_i n_is _is_ is_l s_lo _loc loca ocat cate ated ted_ ed_i d_in _in_ in_G n_Ge _Ger Germ erma rman many any." Note that the input has not been tokenized and there are n-grams spanning token boundaries. We also include non-embedding approaches using bags-of-n-grams within this group as they go beyond word and token representations.

In the following, we explore a subset of bag-of-n-gram models that are used for representation learning, information retrieval and sequence classification tasks.

Representation learning for character sequences. An early study in this category of character-based models is (Schuetze 1992). Its goal was to create corpus-based fixed-length distributed semantic representations for text. To train k-gram embeddings, the top character k-grams are extracted from a corpus along with their co-occurrence counts. Next, singular value decomposition (SVD) is used to create low-dimensional k-gram embeddings, given their cooccurrence matrix. To apply them to a piece of text, the k-grams of the text are extracted and their corresponding embeddings are summed. The study evaluates the k-gram embeddings in the context of word sense disambiguation.

A more recent study (Wieting et al. 2016) trains character n-gram embeddings in an end-to-end fashion with a neural network. They are evaluated on word similarity, sentence similarity and part-of-speech tagging.

Information retrieval. As mentioned before, character n-gram features are widely used in the area of information retrieval (Cavnar 1995; A. Chen et al. 1997; Damashek 1995; De Heer 1974; McNamee and Mayfield 2004; Kettunen et al. 2010).

Sequence classification. Bag-of-n-gram models are used for language identification (Baldwin and Lui 2010; Dunning 1994), topic labeling (Kou et al. 2015), authorship attribution (Peng et al. 2003), word or text similarity (Bojanowski et al. 2017; Eyecioglu and B. Keller 2016; Wieting et al. 2016) and word sense disambiguation (Schuetze 1992).

End-to-end Models

Similar to bag-of-n-gram models, end-to-end models are tokenization-free. Their input is a sequence of characters or bytes and they are directly optimized on a (task-specific) objective. Thus they learn their own task-specific representation of the input sequences. Recently, character-based end-to-end models have gained a lot of popularity because of the success of neural networks.

We explore the subsets of these models that are used for sequence generation, sequence labeling, language modeling and sequence classification tasks.

Sequence-to-sequence generation. In 2011, the authors of (Sutskever, Martens, et al. 2011) proposed an end-to-end model for generating text. They trained RNNs with multiplicative connections on the task of character-level language modeling. Afterwards, they used the model to generate text and found that the model captured linguistic structure and a large vocabulary. It produced only a few uncapitalized non-words and was able to balance parantheses and quotes even over long distances (e.g., 30 characters). A similar study by (Graves 2013) used a long short-term memory network to create character sequences.

Recently, character-based neural network sequence-to-sequence models have been applied to instances of generation tasks like machine translation (Chung, K. Cho, et al. 2016; Kalchbrenner et al. 2016; J. Lee et al. 2016; Y. Wu et al. 2016; Zhen Yang et al. 2016) (which was previously proposed on the token level (Sutskever, Vinyals, et al. 2014)), question answering

(Golub and X. He 2016) and speech recognition (Bahdanau et al. 2016; Chan et al. 2016; Eyben et al. 2009; Graves and Jaitly 2014).

Sequence labeling. Character and character n-gram-based features had already been proposed in 2003 for named entity recognition in an end-to-end manner via a hidden Markov model (Klein et al. 2003). More recently, the authors of (Ma and Hovy 2016) proposed an end-to-end neural network model for named entity recognition and part-of-speech tagging. An end-to-end model has also been suggested for unsupervised, language-independent identification of phrases or words (Gerdjikov and Schulz 2016).

A prominent example of neural end-to-end sequence labeling was recently presented by (Gillick et al. 2016) about multilingual language processing from bytes. A window is slid over the input sequence, which is represented by its byte string. Thus the segments in the window can begin and end mid-word or even mid-character. The authors applied the same model for different languages and evaluated it on part-of-speech tagging and named entity recognition.

Language modeling. Chung, Ahn, et al. in (Chung, Ahn, et al. 2017) proposed a hierarchical multi-scale RNN for language modeling. The model used different timescales to encode temporal dependencies and was able to discover hierarchical structures in a character sequence without explicit tokenization. Other studies on end-to-end language models include (Kalchbrenner et al. 2016; Miyamoto and K. Cho 2016).

Sequence classification. Another recent end-to-end model used character-level inputs for document classification (Y. Xiao and K. Cho 2016; Xiang Zhang and Y. LeCun 2015; Xiang Zhang, J. Zhao, et al. 2015). To capture the long-term dependencies of the input, the authors combined convolutional layers with recurrent layers. The model was evaluated on sentiment analysis, ontology classification, question type classification and news categorization.

1.4 Overview of the dissertation

In a broad definition, human written/oral communications and biological sequences are both instances of 'language'. Many applications in our daily life motivate developing methods for automatic processing of these languages, e.g., machine translation from a foreign language to our mother tongue, DNA processing to identify genetic diseases or for legal purposes, protein engineering for drug design. Within the framework of this dissertation, I develop language processing methods for proteomics, genomics, metagenomics, and human languages. We are interested in the data-driven models making zero or minimal prior assumptions about these languages. Decisions on how to represent the sequential data to the machine (its granularity, whether to consider the order or not, the proper numerical vector representation of sequence elements, etc.) are among the first steps in any data analysis/prediction task. Here, we would like to be data-driven, i.e., we attempt to primarily benefit from the data itself in these very first steps, and try to minimize the use of any prior knowledge. The 'language-agnostic' term in the title refers to this point of view.

Characters and character n-grams (or in bioinformatics terminology, k-mers) have a rich record of being used as representations for language processing applications. Nowadays, character-based models have become increasingly popular in NLP. This development was prompted, especially by the success and popularity of neural networks. Character-level models could overcome the challenge of dealing with tokenization errors and provide the possibility of enhancing or even completely replacing the traditional NLP preprocessing pipeline. Such methods can be vital for bioinformatics research as well, where even naive segmentation of the biological sequences is missing in many cases, and finding a proper representation of the sequences is challenging.

This research is distinct from approaches that have suggested the modeling of biological sequences via formal language theory (Searls 2002; Dong and Searls 1994; Muggleton et al. 2001; Searls 2013). In this dissertation, I focus on recent advances in deep learning and representation learning (Collobert, Weston, et al. 2011; Mikolov, Sutskever, K. Chen, et al. 2013), as well as subsequence-based language processing (Sennrich et al. 2016; Schütze et al. 2016) facilitating a language-agnostic processing of biological and natural language sequences for bioinformatics and NLP tasks, respectively. The projects carried out within the framework of this doctoral research are presented in three main chapters categorized by their the data type, in the areas of proteomics, genomics and metagenomics, and natural languages. A summary of the dissertation organized by task-types is as follows.

- **Task-type 1: Sequence representations and representation learning**
  Finding a proper way to represent data that is interpretable by machines is an important step in any machine learning framework. The conventional method of feature extraction has been to use domain knowledge, which usually requires human intervention and can be expensive, either in time or money and potentially contains the errors that any manual work may suffer from. The advent of deep neural network algorithms allowed automatic encoding of data into a proper representation for downstream machine learning tasks (Bengio et al. 2013). In this dissertation, I propose fixed-length and variable-length sequence representations for proteomics and metagenomics. In addition, I propose a language model-based representation learning for biological sequences. In representation learning, researchers usually distinguish between two categories of evaluations: intrinsic evaluations versus extrinsic evaluations. An intrinsic evaluation examines the quality of the representations (e.g. comparisons of vector similarity with human judgment about the similarity of representation) independent of any specific task, while an extrinsic evaluation tests the strength of the representation in the downstream tasks) (Schnabel et al. 2015). Extrinsic evaluations of the proposed representations are within the framework of the other task-types: e.g., sequence classification, sequence labeling, or marker detection tasks.

- **Task-type 2: Sequence classifications**

Many important bioinformatics and NLP tasks can be categorized as sequence classification. The goal of sequence classification is to assign one or multiple labels out of a finite set of possible labels to the entire sequence. Protein function prediction in proteomics and host phenotype prediction in genomics and metagenomics are examples of such tasks in bioinformatics. In this dissertation, I use the representations proposed in Task-type 1 with machine learning predictive models for sequence classification tasks in proteomics, metagenomics, and natural languages. I explore the use of both classical classifiers (e.g., support vector machine and random forests), as well as the deep neural network as the predictive model.

- **Task-type 3: Sequence labeling**
  Sequence labeling is the task of assigning categorical labels to each element of a sequence. Examples of this type are protein secondary structure prediction, gene finding, protein domain identification, etc. Similar to Task-type 2, I use the representations from Task-type 1 for sequence labeling tasks in proteomics (protein domain identification and secondary structure prediction) and genomics (gene finding). For the protein secondary structure prediction, in particular, I investigate deep learning-based prediction from the protein primary sequence. We study the function of different representations in this task. In addition to the role of features, we evaluate various deep learning architectures including the following models/mechanisms and certain combinations: Bidirectional Long Short-Term Memory (BiLSTM), convolutional neural network (CNN), highway connections, attention mechanism, recurrent neural random fields, and gated multi-scale CNN.

- **Task-type 4: Biomarker and linguistic marker detection and analysis**
  “Biomarkers” is a shortened version of “biological markers”, which refer to a broad range of characteristics that can be objectively identified as indicators of a biological meaning (Strimbu and Tavel 2010). Examples include the genes associated with a certain disease, genotypes identifying the underlying species, protein motifs related to a specific function, etc. Similarly, in linguistics, a marker is a linguistic distinction indicating a grammatical function, such as tense, aspect, and negation markers. In a more broad sense, a marker can be an indicator of any meta-data (e.g., poetic genres (Asgari and Chappelier 2013; Asgari, Ghassemi, et al. 2013), or even the mental health states (Asgari, Nasiriany, et al. 2016)). In this dissertation, I present marker detection methods for both natural language and biological sequences:

  1. **SuperPivot algorithm for marker detection in 1000 natural languages:** For the natural languages, I propose an algorithm called SuperPivot for automatic extraction of linguistic markers in a parallel corpus of 1000 languages with a case study focusing on tense and sentiment.

  2. **DiTaxa for detection of biomarkers from 16S rRNA metagenomics:** For metagenomics, I propose a sub-sequence-based 16S rRNA data processing method

as a new paradigm for sequence phenotype classification and biomarker detection. This method and the related software, called DiTaxa, substitute standard operational taxonomic unit (OTU) clustering or sequence-level analysis by segmenting 16S rRNA reads into the most frequent variable-length subsequences. These subsequences are then used as data representation for downstream phenotype prediction, biomarker detection, and taxonomic analysis.

3. **DiMotif for detection of protein motifs in discriminative fashion:** For proteomics, I propose a general-purpose probabilistic segmentation of protein sequences into commonly occurring variable-length subsequences and then use the representation for discriminative motif mining.

- **Task-type 5: Quantitative comparison of languages**
  In this dissertation, I propose a quantitative distance between genomic settings and natural languages, based on the language model-based embedding presented (described in Task-type 1). For the biological sequences, the embeddings can characterize sequences in terms of the underlying biochemical and biophysical patterns. This property makes the network of biological words (k-mers) or language words in this space an indirect representation of the underlying language model. Considering this fact, I propose a new quantitative measure of the distance between genomic language variations or natural languages based on the divergence between the networks of words in different settings, called word embedding language divergence. I perform a language comparison for the coding regions in the genomes of 12 different organisms (four plants, six animals, and two human subjects). The same method is used for comparing 50 natural languages, where the mapping between words was obtained from aligned nodes in parallel corpora, which was able to cluster similar languages together.

The above mentioned contents are organized by the data-types within the dissertation, **Chapter 2**: proteomics, **Chapter 3**: metagenomics, and **Chapter 4**: human languages. Finally, in **Chapter 5**, I conclude the dissertation and present potential future directions.

Chapter 2

Deep language-agnostic processing of proteomics

2.1 Introduction and chapter overview

Proteins are macromolecules that are crucial elements for the structure and function of cells with a wide array of responsibilities including structural support, intra- and intercellular transport, catalytic activity, defense against bacteria and viruses, muscle contraction, signaling and regulation. Proteins accomplish their diverse functions in interactions with their environments, which can be other macromolecules (such as proteins, DNA, or RNA), chemical compounds, or factors such as the pH or temperature (W. T. Clark and Radivojac 2011; Cooper et al. 2000). Proteins are polymers of small molecules called amino acids,

[¶] The content of this chapter is based on the following publications:

1. Asgari, E., & Mofrad, M. R. (2015). Continuous distributed representation of biological sequences for deep proteomics and genomics. PloS one, 10(11), e0141287.

2. Asgari, E., McHardy, A. C., & Mofrad, M. R. (2019). Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX). Scientific reports, 9(1), 3577.

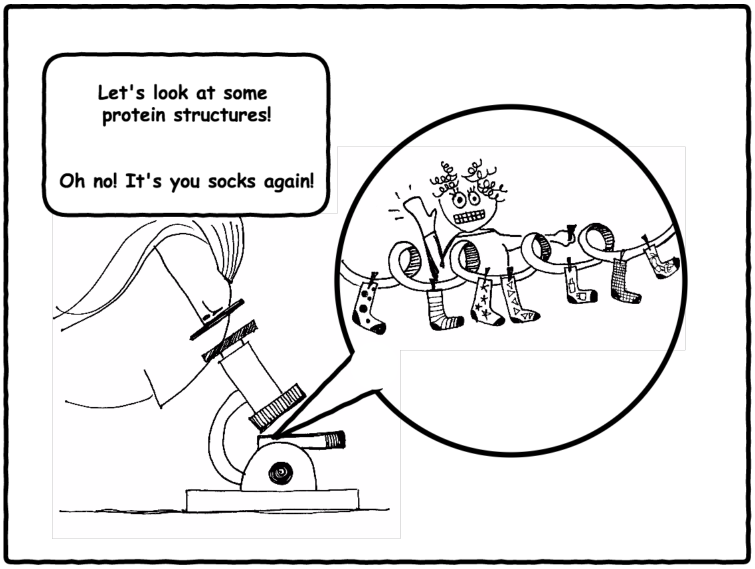
3. Asgari, E., & Mofrad, M. R. (2019). Deep Genomics and Proteomics: Language Model-Based Embedding of Biological Sequences and Their Applications in Bioinformatics. In Leveraging Biomedical and Healthcare Data (pp. 167-181). Academic Press.

4. CAFA challenge report: Zhou, N., Jiang, Y., Bergquist, T. R., Lee, A. J., Kacsoh, B. Z., Crocker, A. W., ... , Asgari, E., Mofrad, M. R, ... , & Davis, L. (2019). The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. bioRxiv, 653105 (Submitted to the Genome Biology Journal).

5. Asgari, E., Poerner, N., McHardy A. C., & Mofrad, M. R. (2019). DeepPrime2Sec: Deep Learning for Protein Secondary Structure Prediction from the Primary Sequences, To be submitted.

of which there are 20 different types, represented by the characters {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}, plus 4 letters indicating ambiguities: B instead of $\{N - or - D\}$, J instead of $\{I - or - L\}$, Z instead of $\{E - or - Q\}$, and X as an completely unknown amino acid. Protein sequences can be as large as chains of 10Ks amino acids[*]; meaning that the space of possible protein sequences is very large. Proteins fold to form a particular three-dimensional structure. It has been proven that the protein's linear sequence can determine their tertiary structures (3D structure) (Cooper et al. 2000). The functions of proteins are highly tied to their 3D structures. Hence, a protein sequence should theoretically hold enough information determining its function. However, finding a mapping from the protein primary sequence to the structure is one of the open challenges in molecular biology (Hunter 1993; Cooper et al. 2000). The large gap between the number of known protein sequences (UniProt database contains 116 million protein sequences to date) and the number of known protein 3D structures (Protein Databank contains only 142K entries for protein 3D structures to date) motivates computational methods and in particular machine learning methods predicting structural and functional information from the protein primary sequences. This is the central goal of this chapter. We focus on developing language processing methods suitable for use of machine learning in the functional and structural annotation of protein sequences using subsequence information as well as deep representation learning.

Similar to any other machine learning framework, performing data analysis on proteomics data requires encoding of protein sequences into vector representations (see 1.2). The conventional method for encoding protein sequences has been the use of biophysical properties in the sequences or direct use of amino acid compositions. Usually, when we use biophysical properties for sequence representation, each amino acid is represented by its biophysical properties, including hydrophobicity, normalized Van der Waals volume, polarity, polarizability, charge, surface tension, secondary structure and solvent accessibility. However, there exist countless combinations of such features to be used in a problem setting. This diversity of features makes a proper features selection very difficult as filtering may require great domain knowledge, which is not always available. One of the main contributions of this dissertation is proposing language processing methods for unsupervised feature extraction from protein sequences. These


Let's look at some protein structures!

Oh no! It's you socks again!

[*] based on sequence lengths on UniProt

methods take advantage of a large number of available protein sequences and subsequently employ the extracted features for down-stream machine learning on structural and functional annotations.

Chapter overview

The contributions of this dissertation in proteomics are in three folds (i) introducing k-mer based distributed representation of protein sequences, (ii) data-driven segmentation of protein sequences into variable-length sub-sequences for protein sequence embedding and motif discovery,and (iii) deep learning for protein secondary structure prediction.

In Section §2.2, we introduce protein-vector (ProtVec) embeddings using the Skip-gram neural networks as an unsupervised representation of protein k-mers, which are eventually used for the representation of the protein sequences using average or summation of k-mer vectors. We evaluate the ProtVec intrinsically and extrinsically. For the intrinsic evaluation, we show that protein vectors are continuous in terms of different biophysical properties. For the extrinsic evaluation, ProtVec is evaluated over classifying protein sequences obtained from Swiss-Prot belonging to 7,027 protein families, where an average family classification accuracy of $94\% \pm 0.03\%$ is obtained outperforming existing family classification methods. In addition, ProtVec is used to distinguish disordered proteins from structured ones. Two databases of disordered sequences are used: the DisProt database as well as a database featuring the disordered regions of nucleoporins rich with phenylalanine-glycine repeats (FG-Nups). Using support vector machine classifiers, FG-Nup sequences are distinguished from structured Protein Data Bank (PDB) sequences with 99% accuracy, and unstructured DisProt sequences from structured DisProt sequences with 100% accuracy. Besides, using GeneVec and ProtVec in intron-exon prediction and protein domain identification tasks could improve the sequence labeling accuracy from 73.84% to 74.99% and from 82.4% to 89.8%, respectively. These results indicate that only providing sequence data, information about protein structure can be determined with high accuracy.

In Section §2.3, we propose peptide-pair encoding (PPE), a general-purpose probabilistic segmentation of protein sequences into commonly occurring variable-length sub-sequences. The idea of PPE segmentation is inspired by the byte-pair encoding (BPE) text compression algorithm, which has recently gained popularity in subword neural machine translation (Sennrich et al. 2016; Kudo 2018). We modify this algorithm by adding a sampling framework allowing for multiple ways of segmenting a sequence. PPE can be inferred over a large set of protein sequences (e.g., Swiss-Prot database) and then applied to a set of unseen sequences. This representation can be widely used as the input to any downstream machine learning tasks in protein bioinformatics. In particular, here, we introduce this representation through protein motif discovery and protein sequence embedding. (i) DiMotif: we present DiMotif as an alignment-free discriminative motif discovery approach and evaluate the method for finding protein motifs in different settings: (1) comparison of DiMotif with two existing approaches on 20 distinct motif discovery problems which are experimentally verified, (2) classification-based approach for the motifs extracted for integrins, integrin-binding proteins, and biofilm formation, and (3) in sequence pattern searching for nuclear localization signal. The DiMotif, in general, obtained high recall scores, while having a comparable F1 score with other methods in the discovery of experimentally verified motifs. Having high recall suggests that the DiMotif can be used for short-list creation for further experimental investigations on motifs. We n the classification-based evaluation, the extracted motifs could reliably detect the integrins, integrin-binding, and biofilm formation-related proteins on a reserved set of sequences with high F1 scores. Motif discovery is a biomarker finding task-type (as defined in 1.4). (ii) ProtVecX: We extend k-mer based protein vector (ProtVec) embedding to variable-length protein embedding using PPE sub-sequences. We show that the new method of embedding can marginally outperform ProtVec in enzyme prediction as well as toxin prediction tasks. In addition, we conclude that the embeddings are beneficial in protein classification tasks when they are combined with raw k-mer features. Finally, in Section §2.5, we conclude with the contributions of this dissertation in proteomics.

In Section §2.4, we introduce deep learning-based prediction of protein secondary structure from the protein primary sequence. We study the function of different features in this task, including one-hot vectors, biophysical features, protein sequence embedding (ProtVec), deep contextualized embedding (known as ELMo), and the Position Specific Scoring Matrix (PSSM). In addition to the role of features, we evaluate various deep learning architectures including the following models/mechanisms and certain combinations: Bidirectional Long Short-Term Memory (BiLSTM), convolutional neural network (CNN), highway connections, attention mechanism, recurrent neural random fields, and gated multi-scale CNN. Our results suggest that PSSM concatenated to one-hot vectors are the most important features for the task of secondary structure prediction. Utilizing the CNN-BiLSTM network, we achieved an accuracy of %69.9 and %70.4 using ensemble top-k models, for 8-class of protein secondary structure on the CB513 dataset, the most challenging dataset for protein secondary structure prediction. Through error analysis on the best performing model, we showed that the misclassification is significantly more common at positions that undergo secondary structure transitions, which is most likely due to the inaccurate assignments of the secondary structure at the boundary regions. Notably, when ignoring amino acids at secondary structure transitions in the evaluation, the accuracy increases to %90.3. Furthermore, the best performing model mostly mistook similar structures for one another, indicating that the deep learning model inferred high-level information on the secondary structure.

2.2 K-mer based protein-vectors (ProtVec) for protein sequence embedding

We introduce a new representation and feature extraction method for biological sequences. Named bio-vectors (BioVec) to refer to biological sequences in general with protein-vectors (ProtVec) for proteins (amino-acid sequences) and gene-vectors (GeneVec) for gene sequences, this representation can be widely used in applications of deep learning in proteomics and genomics. In this section, we focus on protein-vectors that can be utilized in wide array of bioinformatics investigations such as family classification, protein visualization, structure prediction, disordered protein identification, and protein-protein interaction prediction. In this method, we adopt artificial neural network approaches and represent a protein sequence with a single dense n-dimensional vector. To evaluate this method, we apply it in classification of 324,018 protein sequences obtained from Swiss-Prot belonging to 7,027 protein families, where an average family classification accuracy of $93\% \pm 0.06\%$ is obtained, outperforming existing family classification methods. In addition, we use ProtVec representation to predict disordered proteins from structured proteins. Two databases of disordered sequences are used: the DisProt database as well as a database featuring the disordered regions of nucleoporins rich with phenylalanine-glycine repeats (FG-Nups). Using support vector machine classifiers, FG-Nup sequences are distinguished from structured protein sequences found in Protein Data Bank (PDB) with a 99.8% accuracy, and unstructured DisProt sequences are differentiated from structured DisProt sequences with 100.0% accuracy. These results indicate that by only providing sequence data for various proteins into this model, accurate information about protein structure can be determined. Importantly, this model needs to be trained only once and can then be applied to extract a comprehensive set of information regarding proteins of interest. Moreover, this representation can be considered as pre-training for various applications of deep learning in bioinformatics.

The importance of data representation

Can a computer automatically understand a piece of English text to find documents with similar content or automatically translate the given document to French? These types of tasks constitute the area with which Natural Language Processing (NLP) is mainly concerned. The purpose of NLP is to design algorithms allowing computers to understand natural languages for performing specific tasks (e.g., information retrieval, machine translation, semantic analysis, etc.). When we want to discuss a complex concept with an audience unfamiliar with the topic, we model or represent the concept within a framework that is understandable for the audience. The same logic applies in presenting a natural language text to a machine. Computers are experts in dealing with numerical values, vectors, and matrices. Thus, the first step in NLP is to vectorize natural language text for computers. Words are the conventional input units of almost all NLP tasks. Therefore, to utilize machines for language processing we need to find proper vector representations of words that are interpretable by machines.

We expect such representations to preserve some indications of similarity and dissimilarity between words. For instance, when we search a phrase in a search engine we expect the machine to consider words ‘formula’ and ‘equation’ to be similar and consider them dissimilar to an irrelevant word like ‘cuisine’. Thus, we should attribute similar vector representations to the words ‘formula’ and ‘equation’, dissimilar to the vector representation of ‘cuisine’. As a reminder vector similarity can be calculated using operations in linear algebra (e.g. dot product, Euclidian distance, and cosine similarity). Of course semantic similarity is not the only consideration we have in NLP tasks. As an example part-of-speech tagging is one of the routine NLP tasks, where the goal is to label words with their syntactic part-of-speech (e.g. verb, adverb, etc.). Presumably when we want to perform part-of-speech tagging we desire a vector representation incorporating syntactic similarities. The performance of NLP or in general any machine learning task largely depends on the quality of data representation (also know as feature extraction/engineering), in an extend that recently representation learning became an important research area in machine learning (Bengio et al. 2013; Collobert, Weston, et al. 2011). The advent of deep neural network algorithms allowed automatic encoding of data into a proper representation and introduced representation learning as a new field in itself in the realm of machine learning (Bengio et al. 2013). Recent works in the area of representation learning have proposed successful representations of data in computer vision, speech recognition, and natural language processing (Y. LeCun et al. 2015; Graves, Mohamed, et al. 2013; Mikolov, Sutskever, K. Chen, et al. 2013). Similar to the role of textual representations in NLP tasks, representation of biological sequences is key to many bioinformatics tasks, which is facilitated by the recent advances in deep learning (Angermueller et al. 2016).

In this chapter, we propose an unsupervised data-driven distributed representation for biological sequences. This method, called bio-vectors (BioVec) in general and more specifically protein-vectors (ProtVec) for proteins, can be applied to a wide range of problems in bioinformatics, such as protein visualization, protein family classification, structure prediction, domain extraction, and interaction prediction. In this approach, each biological sequence is embedded in an n-dimensional vector that characterizes biophysical and biochemical properties of sequences using neural networks. In the following, we first explain how this method works and how it is trained from 546,790 sequences of Swiss-Prot database. Subsequently, we will analyze the biophysical and the biochemical properties of this representation qualitatively and quantitatively. To further evaluate this feature extraction method, we apply it in classification of 7,027 protein families of 324,018 protein sequences in Swiss-Prot. In the next step, we use this approach for visualization and characterization of two categories of disordered sequences: the DisProt database as well as a database of disordered regions of phenylalanine-glycine nucleoporins (FG-Nups). Finally, we classify these protein families using support vector machine (SVM) classifiers (Cortes and Vapnik 1995). As a key advantage of the proposed method, the embedding needs to be trained only once and then may be used to encode biological sequences in a given problem.

Distributed Representation

Distributed representation has proved one of the most successful approaches in machine learning (Geoffrey E Hinton 1984; Collobert, Weston, et al. 2011; Mikolov, Sutskever, K. Chen, et al. 2013). The main idea in this approach is encoding and storing information about an item within a system through establishing its interactions with other members. Distributed representation was originally inspired by the structure of human memory, where the items are stored in a "content-addressable" fashion. Content-based storing allows for efficiently recalling items from partial descriptions. Since the content-addressable items and their properties are stored within a close proximity, such a system provides a viable infrastructure to generalize features attributed to an item.

Continuous vector representation, as a distributed representation for words, has been recently established in natural language processing (NLP) as an efficient way to represent semantic/syntactic units with many applications. In this model, each word is embedded in a vector in an n-dimensional space. Similar words have close vectors, where similarity is defined in terms of both syntax and semantic. The basic idea behind training such vectors is that the meaning of a word is characterized by its context, i.e. neighboring words. Thus, words and their contexts are considered to be positive training samples. Such vectors can be trained using large amounts of textual data in a variety of ways, e.g. neural network architectures like the Skip-gram model (Mikolov, Sutskever, K. Chen, et al. 2013).

Interesting patterns have been observed by training word vectors using Skip-gram in natural language. Words with similar vector representations show multiple degrees of similarity. For instance, $\overrightarrow{King} - \overrightarrow{Man} + \overrightarrow{Woman}$ resembles the closest vector to the word $\overrightarrow{Queen}$ (Mikolov, K. Chen, et al. 2013).

In this work, we seek unique patterns in biological sequences to facilitate biophysical and biochemical interpretations. We show how Skip-gram can be used to train a distributed representation for biological sequences over a large set of sequences, and establish physical and chemical interpretations for such representations. We propose this as a general-purpose representation for protein sequences that can be used in a wide range of bioinformatics problems, including protein family classification, protein interaction prediction, structure prediction, motif extraction, protein visualization, and domain identification. To illustrate, we specifically tackle visualization and protein family classification problems.

Protein Family Classification

A protein family is a set of proteins that are evolutionarily related, typically involving similar structures or functions. The large gap between the number of known sequences versus the amount of known functional information about sequences has motivated family (function) identification methods based on primary sequences (Enright et al. 2002; Bork et al. 1998; Pedruzzi et al. 2014). Protein Family Database (Pfam) is a widely used source for protein families (Finn et al. 2013). In Pfam, a family can be classified as a "family", "domain", "repeat", or “motif”. In this study, we utilize ProtVec to classify protein families in Swiss-Prot using the information provided by Pfam database and we obtain a high classification accuracy.

Protein family classification based on the primary structures (sequences) has been performed using classifiers such as support vector machine classifier (SVM) (C. Cai et al. 2003; Huynen et al. 2000). Besides the primary sequence, the existing methods typically require extensive information for feature extraction, e.g. hydrophobicity, normalized Van der Waals volume, polarity, polarizability, charge, surface tension, secondary structure and solvent accessibility. The reported accuracies of a previous study on family classification have been in the range of $69.1 - 99.6\%$ for 54 protein families (C. Cai et al. 2003). In another study, researchers used motifs from protein interactions for detecting Structural Classification of Proteins (SCOP) (Murzin et al. 1995) families for 368 proteins, and obtained a classification accuracy of 75% at sensitivity of 10% (Aragues et al. 2007). In contrast, our proposed approach is trained based solely on primary sequence information, yet achieving high accuracy when applied in classifications of protein families.

Disordered Proteins

Proteins can be fully or partially unstructured, i.e. lacking a secondary or ordered tertiary three-dimensional structure. Due to their abundance and the critical roles they play in cell biology, disordered proteins are considered to be an important class of proteins (Dunker et al. 2008). Several studies have focused on disordered peptides and their functional analysis in recent years (Dyson and Wright 2005; Sugase et al. 2007; B. He et al. 2009).

In the present work, we introduce ProtVec for the visualization and characterization of two categories of disordered proteins: DisProt database as well as a database of disordered regions of phenylalanine-glycine nucleoporins (FG-Nups) (Jamali et al. 2011).

DisProt is a database of experimentally identified disordered proteins that categorizes disordered and ordered regions of a collection of proteins (Sickmeier et al. 2007). DisProt Release 6.02 consists of 694 proteins presenting 1539 disordered, and 95 ordered regions. FG-Nups dataset is a collection of FG-Nups disordered sequences (Ando et al. 2013). Nucleoporins form the nuclear pore complex (NPC), the sole gateway for bidirectional transport of cargo between the cytoplasm and the nucleus in eukaryotic cells (Azimi and M. R. Mofrad 2013). Since FG-Nups are mostly computationally identified, only 10 sequences out of 1,138 disordered sequences exist in Swiss-Prot. A recent study on features of FG-Nups versus DisProt showed biophysical differences between FG-Nups and average DisProt sequences (Peyro et al. 2015).

We further propose using protein-vectors for the visualization of biological sequences. Simplicity and biophysical interpretations encoded within ProtVec distinguishes this method from the previous work (Procter et al. 2010; Rutherford et al. 2000). As an example, we use ProtVec for the visualization of FG-Nups, DisProt, and structured PDB proteins. This visualization confirms the results obtained (Peyro et al. 2015) on the biophysical features of FG-Nups and typical disordered proteins. Furthermore, we employ ProtVec to classify FG-Nups versus random PDB sequences as well as DisProt disordered regions versus disport ordered regions.

Protein secondary structure prediction

Protein structure can be described in three main levels: **(i) primary structure** referring to the amino acid sequence, **(ii) secondary structure** referring to the structure of the local segments of the protein categorized into 8 of secondary structures (for each amino acid), and **(ii) tertiary structure** referring to the 3D structure of protein macromolecules. Protein secondary structure prediction can be regarded as sequence labeling task type (described in §1.4). There exist 8 possible categories (Q8 labeling scheme) of secondary structure labels for the structure at each amino acid position: the 3-10 helix (G), alpha helix (H), pi helix (I), turn (T), beta sheet (E), beta bridge (B), bend (S), and loop (L). A simpler labeling scheme is Q3, where the categories are divided into three main classes: helix, strand, and loop/coil. Finding the protein secondary structure is a step toward understanding the protein functions, which is important for many applications including drug design. Here we evaluate the performance of ProtVec features in secondary structure prediction by keeping the same learning algorithm and changing the data representation.

Intron-Exon Prediction

The term genome refers to the sequence of nucleotides that contains the genetic information. Some sections of the genome are functional in the sense that they can be translated to proteins (exonic regions), while other sections cannot be translated to proteins (introns). The rapid decrease in the cost of DNA sequencing and the large amounts of available data discourage us from manually annotating different regions in the genome. In addition, finding all of the patterns that define an intronic or exonic region is not trivial. Thus, this task can widely benefit from data-driven methods.Different methods have been proposed ranging from adhoc automation based on manually extracted patterns to use of Hidden Markov Models, Support Vector Machines, and Conditional Random Fields (Bernal et al. 2007; Krogh 1997; Sonnenburg et al. 2007). Here similar to protein secondary structure prediction, we evaluate the performance of ProtVec features in intron-exon prediction by keeping the same learning algorithm and changing the data representation from one-hot vector representation to ProtVec.

Protein Domain Identification

Domains in proteins refer to regions in the primary sequences that can form a specific tertiary structure. Finding such regions is also an instance of sequence labeling. Protein domain identification is important, as they act as fundamental units of structure and function in a protein (Murzin et al. 1995; Cooper et al. 2000). Here we evaluate ProtVec in tyrosine kinase protein domain identification task, similar to intron-exon and secondary structure prediction tasks. Tyrosine kinase domain is an important domain functioning as a on/off switch for different cellular functions.

Methods

Protein-Space Construction

Our goal is to construct a distributed representation of biological sequences. In the training process of word embedding in NLP, a large corpus of sentences should be fed into the training algorithm to ensure sufficient contexts are observed. Similarly, a large corpus is needed to train distributed representation of biological sequences. We use Swiss-Prot as a rich protein database, which consists of 546,790 manually annotated and reviewed sequences.

The next step in training distributed representations is to break the sequences into sub sequences (i.e. biological words). The simplest and most common technique in bioinformatics to study sequences involves fixed-length overlapping k-mers (Ganapathiraju et al. 2002; Srinivasan et al. 2013; Vries and X. Liu 2008). However, instead of using k-mers directly in feature extraction, we utilize k-mer modeling for training a general purpose distributed representation of sequences. This so-called embedding model needs to be trained only once and may then be adopted in feature extraction part of specific problems.

In k-mer modeling of protein informatics, usually an overlapping window of 3 to 6 residues is used. Instead of taking overlapping windows, we generate 3 lists of shifted non-overlapping words, as shown in Fig. 2.1. Evaluating K-nearest neighbors in a 2xfold cross-validation for different window sizes, embedding vector sizes and overlapping versus non-overlapping k-mers showed a more consistent embedding training for a window size of 3 and the mentioned splitting.

Original Sequence

$$^{(1)}\overrightarrow{M}^{(2)}\overrightarrow{A}^{(3)}\overrightarrow{F}SAEDVLKEYDRRRRMEAL..$$

Splittings

$$\begin{cases} 1) & \text{MAF, SAE, DVL, KEY, DRR, RRM, ..} \\ 2) & \text{AFS, AED, VLK, EYD, RRR, RME, ..} \\ 3) & \text{FSA ,EDV, LKE, YDR, RRR, MEA, ..} \end{cases}$$

Figure 2.1. Protein sequence splitting. In order to prepare the training data, each protein sequence will be represented as three sequences (1,2,3) of 3-mers.

The same procedure is applied on all 546,790 sequences in Swiss-Prot, thus at the end we obtain a corpus consisting of $546,790 \times 3 = 1,640,370$ sequences of 3-mers (3-mer is a "biological" word consisting of 3 amino acids). The next step is training the embedding based on such data through a Skip-gram neural network. In training word vector representations, Skip-gram attempts to maximize the probability of observed word sequences (contexts). In other words, for a given training sequence of words we would like to find their corresponding n-dimensional vectors maximizing the following average log probability function. Such a constraint allows similar words to assume a similar representation in this space.

$$\begin{aligned} &\underset{v,v}{\text{argmax}} \frac{1}{N} \sum_{i=1}^{N} \sum_{-c\leq j\leq c, j\neq 0} \log p(w_{i+j}|w_i) \\ &p(w_{i+j}|w_i) = \frac{\exp\left(v'^{T}_{w_{i+j}} v_{w_i}\right)}{\sum_{k=1}^{W} \exp\left(v'^{T}_{w_k} v_{w_i}\right)}, \end{aligned} \tag{2.1}$$

where N is the length of the training sequence, $2c$ is the window size we consider as the context, w_i is the center of the window, W is the number of words in the dictionary and v_w and v'_w are input and output n-dimensional representations of word w, respectively. The probability $p(w_{i+j}|w_i)$ is defined using a softmax function. Hierarchical softmax or negative sampling are efficient approximations of such a softmax function. In the implementation we use (Word2Vec) (Mikolov, Sutskever, K. Chen, et al. 2013) negative sampling has been utilized, which is considered as the state-of-the-art for training word vector representation. Negative sampling uses the following objective function in the calculation of the word vectors:

$$\arg\max_{\theta} \prod_{(w,c)\in D} p(D=1|c,w;\theta) \prod_{(w,c)\in D'} p(D=0|c,w;\theta), \tag{2.2}$$

where D is the set of all word and context pairs (w,c) existing in the training data (positive samples) and D' is a randomly generated set of incorrect (w,c) pairs (negative samples).

$p(D=1|w,c;\theta)$ is the probability that (w,c) pair came from the training data and $p(D=0|w,c;\theta)$ is the probability that (w,c) did not come from the training data. The term $p(D=1|c,w;\theta)$ can be defined using a sigmoid function on the word vectors:

$$p(D=1|w,c;\theta) = \frac{1}{1+e^{-v_c \cdot v_w}},$$

where the parameters θ are the word vectors we train within the optimization framework: v_c and $v_w \in R^d$ are vector representations for the context c and the word w respectively. In Equation 2.2, the positive samples maximize the probabilities of the observed (w,c) pairs in the training data, while the negative samples prevent all vectors from having the same value by disallowing some incorrect (w,c) pairs. To train the embedding vectors, we consider a vector size of 100 and a context size of 25. Thus each 3-mer is presented as a vector of size 100.

Protein-Space Analysis

To qualitatively analyze the distribution of various biophysical and biochemical properties within the training space, we project all 3-mer embeddings from 100-dimensional space to a 2D space using Stochastic Neighbor Embedding (Maaten and G. Hinton 2008). Mass, volume, polarity, hydrophobicity, charge, and van der Waals volume properties were analyzed. The data is adopted from (McGregor 2004). In addition, to quantitatively measure the continuity of these properties in the protein-space, the best Lipschitz constant, i.e. the smallest k satisfying is calculated:

$$d_f(f_{prop}(w_1), f_{prop}(w_2)) \leq k \times d_w(w_1, w_2), \tag{2.3}$$

where f is the scale of one of the properties of a given 3-mers (e.g., average mass, hydrophobicity, etc.), d is the distance metric, d_f is the absolute value of score differences, and d_w is Euclidian distance between two 3-mers w_1 and w_2. The Lipschitz constant is calculated for the aforementioned properties.

Protein Family Classification

To evaluate the strength of the proposed representation, we set up a classification task on protein families. Family information of 324,018 protein sequences in Swiss-Prot is extracted from the Protein Family (Pfam) database, resulting in a total of 7,027 distinct families for Swiss-Prot sequences. Each sequence is represented as the summation of the vector representation of overlapping 3-mers, similar to obtaining of the sentence embedding from the word embeddings in natural languages (Asgari and Sanaei 2017). Thus, each sequence is presented as a vector of size 100. For each family type, the same number of instances from Swiss-Prot are selected randomly to form the negative examples. Support vector machine classifiers are used to evaluate the strength of ProtVec in the classification of protein families through 10×fold cross-validations. We perform the classification over 7,027 protein families consisting of 324,018 sequences. For the evaluation we report specificity (true negative rate), sensitivity (true positive rate), and the accuracy of family classifications.

$$Sensitivity = TP\ rate = \frac{TP}{TP + FN}$$

$$Specificity = 1 - FP\ rate = \frac{TN}{FP + TN}$$

$$Accuracy = \frac{TP + TN}{P + N}$$

Visualization and Classification of Disordered Proteins

Two databases of disordered proteins are used for disordered protein prediction: DisProt database (694 sequences) and FG-Nups dataset (1,138 sequences).

FG-Nups Characterization

To distinguish the characteristics of FG-Nups, a collection of 1,138 FG-Nups and two random sets of 1,138 structured proteins from Protein Data Bank (PDB)() are compared. Since PDB sequences on average have a shorter length than disordered proteins, the two sets are selected from PDB in such a way that they have an average length of 900 residues, the same as the average length of the disordered protein dataset. For visualization purposes, the ProtVec is reduced from 100 dimensions to a 2D space using Stochastic Neighbor Embedding (Maaten and G. Hinton 2008).

We quantitatively evaluate how ProtVec can be used to distinguish between FG-Nups versus typical PDB sequences using a support vector machine binary classifier. The positive examples were the aforementioned 1,138 disordered FG-Nups proteins and the negative examples (again 1,138 sequences) are selected randomly from PDB with the same average length of disordered sequences ($\approx$ 900 residues). We present each protein sequence as a summation of its ProtVecs of all 3-mers. Since the average length of structured proteins is shorter than FG-Nups, and to avoid trivial cases, the PDB sequences are selected in a way to maintain the same average length.

DisProt Characterization

To distinguish the characteristics of DisProt sequences, we use DisProt Release 6.02, consisting of 694 proteins presenting 1539 disordered and 95 ordered regions, and perform the same experiment as for FG-Nups with DisProt sequences.

Sequence labeling tasks

Variety of statistical learning methods for sequence labeling and more generally structured prediction have been proposed in the past decade in natural language processing, computer vision, and bioinformatics (Lafferty et al. 2001; Nowozin and Lampert 2011; Taskar et al. 2005; Daume 2006; Z. Wang et al. 2011) including recent neural sequence labeling methods benefiting from combination of bidirectional recurrent neural network and conditional random fields (CRF) (Lample et al. 2016), which in some cases are also combined with convolutional neural networks (Reimers and Gurevych 2017). However, in this section for simplicity we mainly focus on max-margin Markov network (M^3Net) sequence labeling model (Taskar et al. 2005) as the emphasis is on the representation and not over the learning algorithm. M^3Net as one of the successful statistical approaches for structured prediction in machine learning. The reason behind this choice is to benefit from the strength of graphical models in modeling the dependencies, as well as the ability of max-margin Markov networks in incorporation of kernels to deal with high-dimensional features, where we can use ProtVec embeddings. We use the PyStruct implementation (A. C. Mueller and Behnke 2014) of M^3Net, which uses Block-Coordinate Frank-Wolfe optimization (Lacoste-Julien et al. 2012).

Intron-Exon Prediction: we perform intron-exon prediction using max-margin Markov networks to evaluate performance of bio-vector data representation versus one-hot vector representation of nucleotides. We use a dataset of 3000 DNA sequences with marked intron and exon regions from the exon–intron database (Shepelev and Fedorov 2006). We use 80% of the sequences for training and 20% for evaluation. The embedding are trained similar to ProtVec over all sequences exist in the Shepelev database (Shepelev and Fedorov 2006) over 6-mers. For each position in the sequence either one-hot vector representation or average of the 6-mers bio-vectors having overlaps with that positions are used. The label for each position is either 0 (intron) or 1 (exon). At the end we report accuracy of labeling for both one-hot vector representation and bio-vectors.

Domain identification: we use a dataset of 2340 protein sequences with the tyrosine kinase domain extracted from the SCOP database (Murzin et al. 1995). We use 80% of the sequences for training and 20% for evaluating of the use of bio-vectors data representation versus one-hot vectors in M^3Net. Each position in the input sequence is presented as the average of the vector representation for the previous and the next 3-mer segment. The label for each position is either 0 (not tyrosine kinase domain) or 1 (tyrosine kinase domain). At the end we report accuracy of labeling for both one-hot vector representation and ProtVec.

Secondary structure prediction: for protein secondary structure prediction standard dataset cullPDB/cb513 consists of 5534/513 (train/test) sequences labeled with secondary structure (Jian Zhou and Troyanskaya 2014) is used. Similar to intron-exon prediction at each position, we take the average of all 3-mer vectors that position exist in. Each position in the input sequence is presented as the average of the vector representation for the previous and the next 3-mer segment. The label for each position is either selected from Q8 or Q3 schemes (described in section §2.2). At the end we report accuracy of labeling for both one-hot vector representation and ProtVec. Later in §2.4, we provide a more comprehensive study on this task.

Results

Protein-Space Analysis

Although the protein-space is trained based on only the primary sequences of proteins, it offers several interesting biochemical and biophysical implications. In order to study these features, we visualized the distribution of different criteria, including mass, volume, polarity, hydrophobicity, charge, and van der Waals volume in this space. To do so, for each 3-mer we conducted qualitative and quantitative analyses as described below.

Qualitative Analysis: In order to visualize the distribution of the aforementioned properties, we projected all 3-mer embeddings from 100-dimensional space to a 2D space using Stochastic Neighbor Embedding (t-SNE) (Maaten and G. Hinton 2008). In the diagrams presented in Fig. 2.2, each point represents a 3-mer and is colored according to its scale in each property. Interestingly, as can be seen in the figure, 3-mers with the same biophysical and biochemical properties were grouped together. This observation suggests that the proposed embedding not only encodes protein sequences in an efficient way that proved useful for classification purposes, but also reveals some important physical and chemical patterns in protein sequences.

Quantitative Analysis: Although Fig. 2.2 illustrates the smoothness of protein-space with respect to different physical and chemical meanings, we required a quantitative approach to measure the continuity of these properties in the protein-space. To do so, we calculated

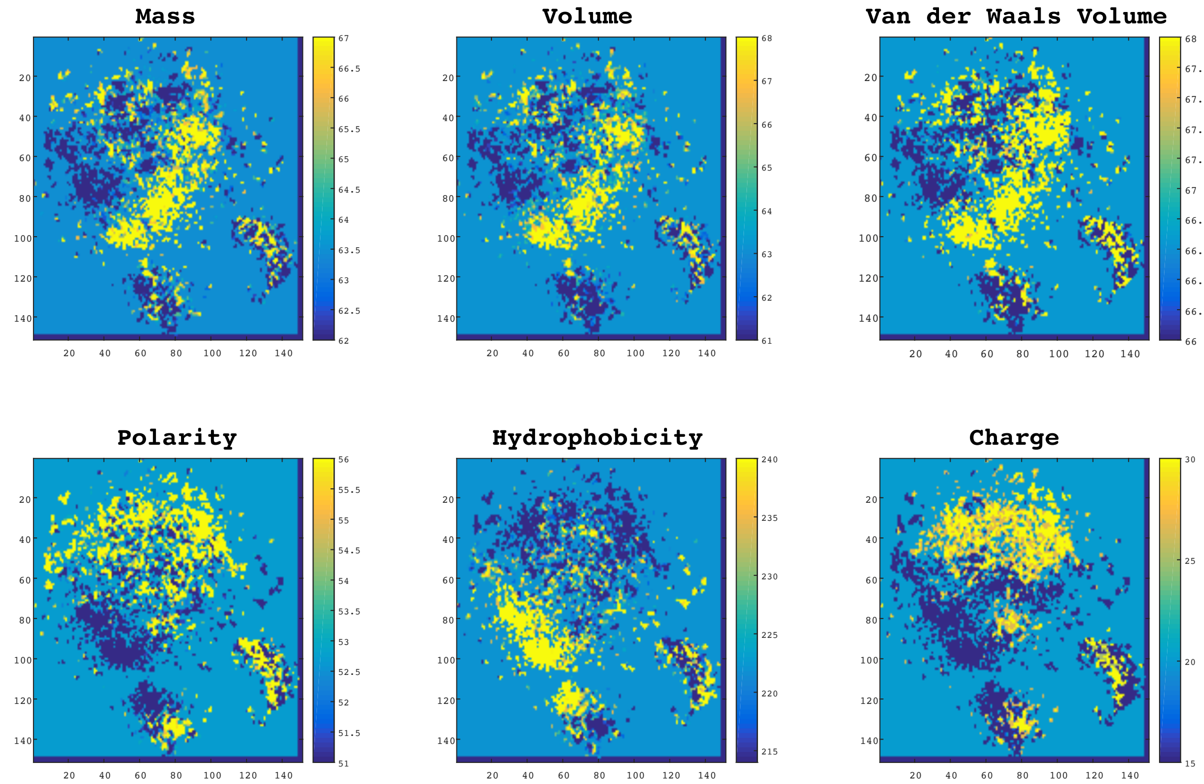


Figure 2.2. Normalized distributions of biochemical and biophysical properties in protein-space. In these plots, each point represents a 3-mer (a word of three residues) and the colors indicate the scale for each property. Data points in these plots are projected from a 100-dimensional space a 2D space using t-SNE. As it is shown words with similar properties are automatically clustered together meaning that the properties are smoothly distributed in this space.

the best Lipschitz constant. For all 6 properties presented in Fig. 2.2, we calculated the minimum k. To evaluate this result we made an artificial space called "scrambled space" by randomly shuffling the labels of 3-mers in the 100 dimensional space. Table 2.1 contains the values of Libschitz constants for protein-space versus the "scrambled space" with respect to different properties and also their ratio.

Normally if $k = 1$ the function is called a short map, and if $0 \leq k < 1$ the function is called a contraction. The results suggest that the protein-space is on average 2-times smoother in terms of physical and chemical properties than a random space. This quantitative result supports our qualitative observation of the space structure in Fig.2.2, and suggests that our training space encodes, 3-mers in an informative manner.

Protein Family Classification

In order to evaluate the strength of ProtVec, we performed classifications of 7,027 protein families and obtained a weighted average accuracy of $93 \pm 0.06\%$, which exhibits a more reliable result than the existing methods. In contrast to the existing methods, our proposed approach is trained based on primary sequence information alone.

Table 2.2 shows the sensitivity, specificity, and the accuracy for the most frequent families in Swiss-Prot. These results suggest that structural features of proteins can be accurately

Table 2.1. Using Lipschitz number to evaluate the continuity of ProtVec with respect to biophysical and biochemical properties

	Lipschitz Number		
Property	protein-Space	The scrambled space	Ratio
Mass	0.3137	0.6605	0.4750
Volume	0.3742	0.6699	0.5586
Van Der Waal Volume	0.3629	0.6431	0.5643
Polarity	0.4757	1.2551	0.3790
Hydrophobicity	0.608	1.448	0.4203
Charge	0.8733	1.3620	0.6412
Average	0.50	1.01	0.51

predicted from the primary sequence information solely. The average accuracy for the first 1,000 (261,149 sequences), 2,000 (293,957 sequences), 3,000 (308,292 sequences), and 4,000 (316,135 sequences) frequent families were respectively 94%±0.05%,93%±0.05%, 92%±0.06%, and 91% ± 0.08%. To compute the overall accuracy for all 7,026 families, we calculated the weighted average accuracy, because for the families with number of instances less than 10, the validation set are not statistically sufficient and they should have less contribution in the overall accuracy. The weighted accuracy of all 7,027 families (weighted based on the number of instances) was 93% ± 0.06%.

Disordered Proteins Visualization and Classification

Due to the functional importance of disordered proteins, prediction of unstructured regions of disordered proteins and determining the sequence patterns featured in disordered regions is a critical problem in protein bioinformatics. We evaluated the ability of ProtVec to characterize and discern disordered protein sequences from structured sequences.

FG-Nups Characterization: In this case study, we used the FG-Nups collection of 1,138 disordered proteins containing disorder regions with a fraction of at least one third of the sequence length. For comparison purposes, we also collected two sets of structured proteins from Protein Data Bank (PDB).

In order to visualize each dataset, we reduced the dimensionality of the protein-space using Stochastic Neighbor Embedding (Maaten and G. Hinton 2008; Platzer 2013) and then generated the 2D histogram of all overlapping 3-mers occurring in each dataset. As shown in Fig. 2.3 (see column (b)), the two random sets from structured proteins had nearly identical patterns. However, the FG-Nups dataset exhibits a substantially different pattern. To amplify the characteristic of disordered sequences we have also examined the histogram of disordered regions of FG-Nups (see Fig. 2.3, column (a)).

In the next step, we quantitatively evaluated how ProtVec can be used to distinguish between FG-Nups versus typical PDB sequences using a support vector machine binary classification. The positive examples were the above mentioned 1,138 disordered FG-Nups proteins and negative examples (again 1,138 sequences) were selected randomly from PDB with the same average length of disordered sequences (≈ 900 residues). We represented each protein sequence as a summation of its ProtVecs of all 3-mers. Since on average the length of structured proteins were shorter than FG-Nups, in order to avoid trivial cases, the PDB sequences were selected in such a way as to maintain the same average length. But still, an accuracy of 99.81% was obtained with high sensitivity and specificity (Table 2.3). The distribution of the classified proteins in a 2D space is shown in Fig. 2.4.

DisProt characterization: In this part, we used DisProt consisting of 694 proteins presenting 1539 disordered, and 95 ordered regions. We performed the same analysis as we did for FG-Nups with DisProt sequences (see Fig. 2.3 column (c)). Since the size of DisProt was relatively small compared to that of the FG-Nups, the scales of columns (a),(b) were not comparable with column (c) (see Fig. 2.3). The visualization of disordered regions of DisProt sequences (Fig. 2.3 column (c), on top) revealed a different characteristic than FG-Nups disordered regions (Fig. 2.3 column (a), on top). A visual comparison between Fig. 2.3 and 2.2 suggests that the FG-Nups have a significantly higher amount of hydrophobic residues and less polar residues in their disordered regions than the experimentally identified disordered proteins in DisProt (Peyro et al. 2015; Ando et al. 2013). Additionally, the DisProt disordered regions versus DisProt ordered regions can be classified with 100% accuracy respectively using SVM and ProtVec.

Sequence labeling performances

The results of the experiment for different sequence labeling tasks are provided in Table 2.4. As the results show incorporating the embedding features can increase the accuracy of sequence labeling in intron-exon prediction and protein domain identification tasks. However, embedding could not improve the secondary structure prediction performance. Surprisingly, there was no significant difference in the precision and the recall for use of one-hot vector representation and the ProtVec. The results are also far from the state-of-the-art performance in secondary structure prediction (Johansen et al. 2017; Jiyun Zhou et al. 2018). Thus, for the secondary structure prediction, changing the model and the representation might help in seeing the representation differences. In Section§2.4 we concentrate on the secondary structure task and investigate different representations and deep learning architectures.

Conclusions

An unsupervised data-driven distributed representation, called ProtVec, was proposed for application of machine learning approaches in biological sequences. By training this representation solely on protein sequences, our feature extraction approach was able to capture a diverse range of meaningful physical and chemical properties. We demonstrated that ProtVec

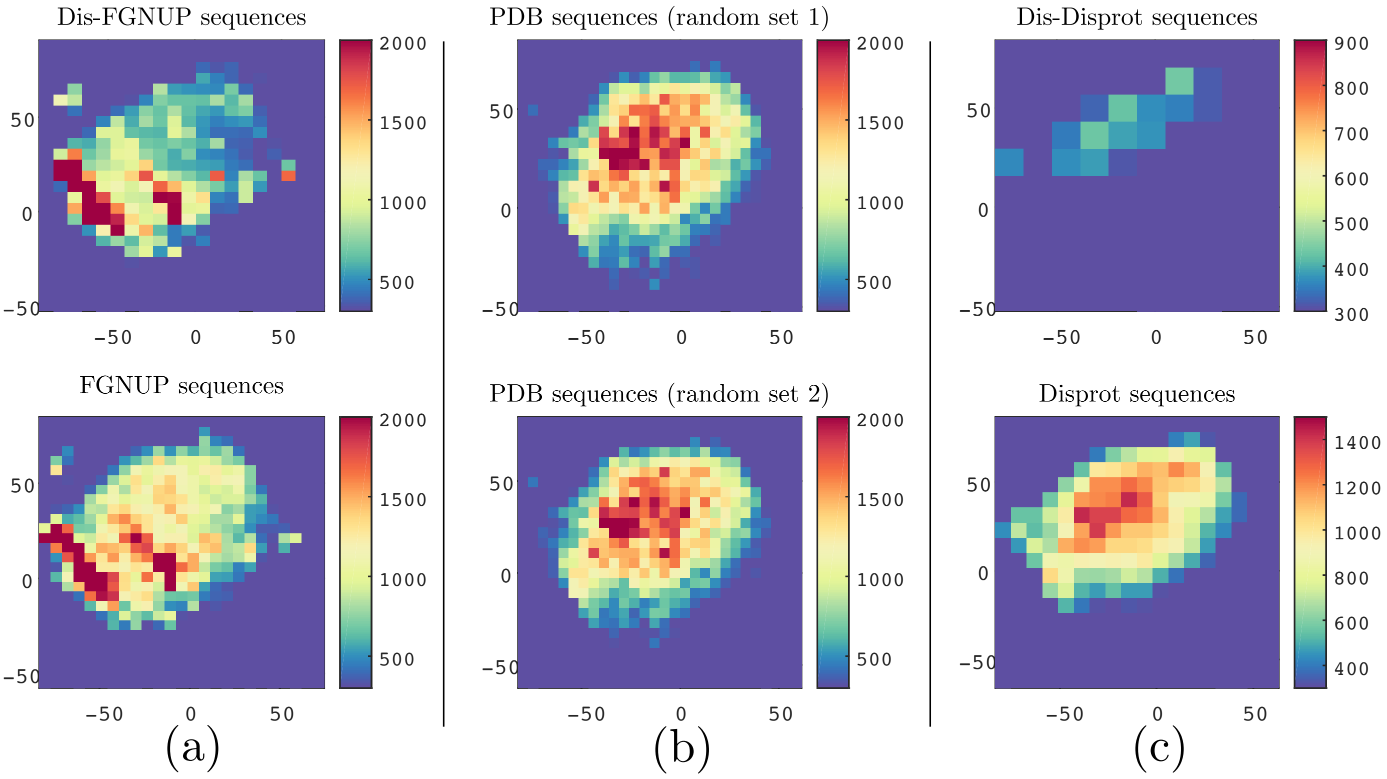


Figure 2.3. Visualization of protein sequences using ProtVec can characterize FGNUPs versus Disport disordered sequences and structured sequences. Column (a) compares FG Nup sequences 2D histogram (at the bottom) with 2D histogram of FG Nup disordered regions (on top). Column (b) compares 2D histogram two random sets of structured sequences with the same average length as the FG-Nups. Column (c) compares between 2D histogram of DisProt sequences (at the bottom) and 2D histogram of DisProt disordered regions (on top)

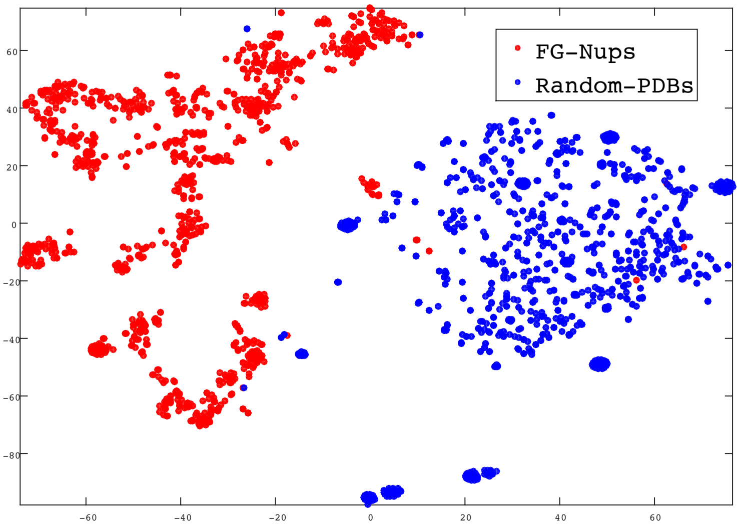


Figure 2.4. Classification of FG-Nups versus PDB structured sequences. In this figure, each point presents a protein projected into a 2D space.

can be used as an informative and dense representation for biological sequences in protein family classification, and obtained an average family classification accuracy of 93%.

We further proposed ProtVec as a powerful approach for protein data visualization and showed the utility of this approach by providing an example in characterization of disordered protein sequences vs. structured protein sequences. Our results suggest that ProtVec can characterize protein sequences in terms of biochemical and biophysical interpretations of the underlying patterns. In addition, this dense representation of sequences can help to discriminate between various categories of sequences, e.g. disordered proteins. Furthermore, we demonstrated that ProtVec was able to identify disordered sequences with an accuracy of nearly 100%. In addition, incorporation of bio-vector representation versus one-hot vector features in Max- margin Markov Network (M^3Net) for intron-exon prediction and protein domain identification tasks could improve the sequence labeling accuracy from 73.84% to 74.99% and from 82.4% to 89.8%, respectively.

Another advantage of this method is that embeddings could be trained once and then used to encode biological sequences in any given problem. In general, machine learning approaches in bioinformatics can widely benefit from bio-vectors (ProtVec and GeneVec) representation. This representation can be considered as pre-training for various applications of deep learning in bioinformatics. In particular, ProtVec can be used in protein interaction predictions, structure prediction, and protein data visualization.

Table 2.2. Performance of protein family classification using SVM and ProtVec over some of the most frequent families in Swiss-Prot. Families are sorted with respect to their frequency in Swiss-Prot.

	Training instances		Classification Result		
Family name	# of positive sequences	# of negative sequences	Specificity	Sensitivity	Accuracy
50S ribosome-binding GTPase	3,084	3,084	0.95	0.93	0.94
Helicase conserved C-terminal domain	2,518	2,518	0.83	0.80	0.82
ATP synthase alpha-beta family, nucleotide-binding domain	2,387	2,387	0.98	0.97	0.97
7 transmembrane receptor (rhodopsin family)	1,820	1,820	0.95	0.96	0.95
Amino acid kinase family	1,750	1,750	0.91	0.92	0.91
ATPase family associated with various cellular activities (AAA)	1711	1711	0.92	0.90	0.91
tRNA synthetases class I (I, L, M and V)	1,634	1,634	0.97	0.97	0.97
tRNA synthetases class II (D, K and N)	1,419	1,419	0.88	0.83	0.85
Major Facilitator Superfamily	1,303	1,303	0.95	0.97	0.96
Hsp70 protein	1,272	1,272	0.97	0.97	0.97
NADH-Ubiquinone-plastoquinone (complex I), various chains	1,251	1,251	0.97	0.97	0.97
Histidine biosynthesis protein	1,248	1,248	0.96	0.97	0.97
TCP-1-cpn60 chaperonin family	1,246	1,246	0.95	0.96	0.95
EPSP synthase (3-phosphoshikimate 1-carboxyvinyltransferase)	1,207	1,207	0.96	0.96	0.96
Aldehyde dehydrogenase family	1,200	1,200	0.93	0.94	0.94
Shikimate - quinate 5-dehydrogenase	1,128	1,128	0.87	0.89	0.88
GHMP kinases N terminal domain	1,120	1,120	0.88	0.92	0.90
Ribosomal protein S2	1,083	1,083	0.95	0.96	0.95
Ribosomal protein S4-S9 N-terminal domain	1,072	1,072	0.95	0.97	0.96
Ribosomal protein L16p-L10e	1,053	1,053	0.95	0.96	0.96
KOW motif	1,047	1,047	0.93	0.95	0.94
Uncharacterized protein family UPF0004	1,044	1,044	0.95	0.97	0.96
Ribosomal protein S12-S23	1,016	1,016	0.94	0.98	0.96
GHMP kinases C terminal	1,011	1,011	0.88	0.92	0.90
Ribosomal protein S14p-S29e	997	997	0.93	0.98	0.95
Ribosomal protein S11	980	980	0.96	0.98	0.97
UvrB-uvrC motif	968	968	0.94	0.96	0.95
Ribosomal protein L33	958	958	0.96	0.98	0.97
BRCA1 C Terminus (BRCT) domain	956	956	0.94	0.95	0.95
RF-1 domain	950	950	0.93	0.97	0.95
Ankyrin repeats (3 copies)	944	944	0.89	0.88	0.88
Ribosomal protein L20	932	932	0.96	0.99	0.97
RNA polymerase beta subunit	912	912	0.94	0.97	0.95
Ribosomal protein S18	908	908	0.93	0.97	0.95
ATP synthase B-B CF(0)	900	900	0.92	0.94	0.93
Peptidase family M20-M25-M40	889	889	0.92	0.93	0.93
Ribosomal protein L18e-L15	887	887	0.93	0.96	0.95
Glucose inhibited division protein A	886	886	0.95	0.96	0.95
NADH-ubiquinone-plastoquinone oxidoreductase chain 4L	885	885	0.94	0.97	0.96
lactate-malate dehydrogenase, NAD binding domain	880	880	0.92	0.94	0.93
HD domain	879	879	0.93	0.93	0.93
Ribosomal protein S10p-S20e	873	873	0.95	0.97	0.96
Pyridoxal-phosphate dependent enzyme	870	870	0.91	0.91	0.91
Ribosomal L18p-L5e family	860	860	0.93	0.96	0.94
Ribosomal protein L3	855	855	0.94	0.97	0.96
tRNA synthetases class I (M)	843	843	0.94	0.96	0.95
UbiA prenyltransferase family	841	841	0.94	0.95	0.95
Ribosomal protein L4-L1 family	841	841	0.94	0.95	0.95
Ribosomal protein S16	840	840	0.93	0.97	0.95
Ribosomal protein S13-S18	840	840	0.94	0.97	0.95
MraW methylase family	837	837	0.95	0.98	0.96
Ribosomal L32p protein family	825	825	0.94	0.97	0.95
Elongation factor TS	819	819	0.94	0.97	0.96
Tetrahydrofolate dehydrogenase-cyclohydrolase, catalytic domain	817	817	0.94	0.96	0.95
ATP synthase delta (OSCP) subunit	813	813	0.93	0.96	0.94
tRNA synthetases class I (C) catalytic domain	812	812	0.95	0.97	0.96
SecA Wing and Scaffold domain	805	805	0.95	0.97	0.96
Ribonuclease HII	795	795	0.93	0.94	0.93
Ribosomal protein L31	795	795	0.97	0.99	0.98
Ribosomal L27 protein	794	794	0.98	0.99	0.99
IPP transferase	794	794	0.93	0.95	0.94
GTP-binding protein LepA C-terminus	793	793	0.96	0.98	0.97
Ribosomal protein L17	791	791	0.92	0.96	0.94
Ribosomal protein L23	790	790	0.91	0.96	0.94
Ribosomal protein L10	781	781	0.90	0.92	0.91
Ribosomal protein L19	780	780	0.94	0.97	0.95
Ribosomal protein S20	774	774	0.95	0.97	0.96
Ribosomal protein L35	769	769	0.93	0.97	0.95
Phosphoglucomutase-phosphomannomutase, C-terminal domain	768	768	0.92	0.96	0.94
AMP-binding enzyme	767	767	0.87	0.89	0.88
Ribosomal prokaryotic L21 protein	766	766	0.93	0.96	0.95
tRNA methyl transferase	759	759	0.94	0.96	0.95
Ribosomal L29 protein	757	757	0.95	0.97	0.96
Glycosyl transferase family, a-b domain	754	754	0.90	0.91	0.91
Translation initiation factor IF-2, N-terminal region	750	750	0.96	0.98	0.97
Ribosomal L28 family	749	749	0.93	0.98	0.95
Glycosyl transferase family 4	739	739	0.96	0.98	0.97
tRNA synthetases class I (R)	736	736	0.93	0.96	0.95
Bacterial trigger factor protein (TF) C-terminus	733	733	0.95	0.96	0.95
For the first 1,000 families	261,149	261,149	0.92	0.95	0.94
For the first 2,000 families	293,957	293,957	0.90	0.96	0.93
For the first 3,000 families	308,292	308,292	0.89	0.96	0.92
For the first 4,000 families	316,135	316,135	0.87	0.96	0.91
Weighted average for all 7,027 families	324,018	324,018	0.91	0.95	0.93

Table 2.3. The performance of FG-Nups disordered protein classification in a 10xFold cross-validation using SVM

Sensitivity	Specificity	Accuracy
0.9987	0.9974	0.9981

Table 2.4. The accuracy of sequence labeling task in intron-exon prediction, domain identification, and secondary structure prediction

Task	M^3Net / **One hot**	M^3Net / **Embedding**
Gene Prediction	73.84%	74.99%
Domain Identification	82.4%	89.8%
Secondary Structure (Q8)	40.0%	40.0%
Secondary Structure (Q3)	56.0%	56.0%

2.3 Variable-length protein segmentation for motif mining and sequence embedding

In this section, we present peptide-pair encoding (PPE), a general-purpose probabilistic segmentation of protein sequences into commonly occurring variable-length sub-sequences. The idea of PPE segmentation is inspired by the byte-pair encoding (BPE) text compression algorithm, which has recently gained popularity in subword neural machine translation. We modify this algorithm by adding a sampling framework allowing for multiple ways of segmenting a sequence. PPE segmentation steps can be learned over a large set of protein sequences (Swiss-Prot) or even a domain-specific dataset and then applied to a set of unseen sequences. This representation can be widely used as the input to any downstream machine learning tasks in protein bioinformatics. In particular, here, we introduce this representation through protein motif discovery and protein sequence embedding. (i) DiMotif: we present DiMotif as an alignment-free discriminative motif discovery method and evaluate the method for finding protein motifs in three different settings: (1) comparison of DiMotif with two existing approaches on 20 distinct motif discovery problems which are experimentally verified, (2) classification-based approach for the motifs extracted for integrins, integrin-binding proteins, and biofilm formation, and (3) in sequence pattern searching for nuclear localization signal. The DiMotif, in general, obtained high recall scores, while having a comparable F1 score with other methods in the discovery of experimentally verified motifs. Having high recall suggests that the DiMotif can be used for short-list creation for further experimental investigations on motifs. In the classification-based evaluation, the extracted motifs could reliably detect the integrins, integrin-binding, and biofilm formation-related proteins on a reserved set of sequences with high F1 scores. (ii) ProtVecX: we extend k-mer based protein vector (ProtVec) embedding to variable-length protein embedding using PPE sub-sequences. We show that the new method of embedding can marginally outperform ProtVec in enzyme prediction as well as toxin prediction tasks. In addition, we conclude that the embeddings are beneficial in protein classification tasks when they are combined with raw k-mer features.

Tokenization in Human Language Processing versus Protein Informatics

Bioinformatics and natural language processing (NLP) are research areas that have greatly benefited from each other since their beginnings and there have been always methodological exchanges between them. Levenshtein distance (Levenshtein 1966) and Smith–Waterman (Waterman et al. 1976) algorithms for calculating string or sequence distances, the use of formal languages for expressing biological sequences (Searls 1993; Searls 2002), training language model-based embeddings for biological sequences (Asgari and M. R. Mofrad 2015), and using state-of-the-art neural named entity recognition architecture (Lample et al. 2016) for secondary structure prediction (Johansen et al. 2017) are some instances of such influences. Similar to the complex syntax and semantic structures of natural languages, certain biophysical and biochemical grammars dictate the formation of biological sequences. This assumption has motivated a line of research in bioinformatics to develop and adopt language processing methods to gain a deeper understanding of how functions and information are encoded within biological sequences (Yandell and Majoros 2002; Searls 2002; Asgari and M. R. Mofrad 2015). However, one of the apparent differences between biological sequences and many natural languages is that biological sequences (DNA, RNA, and proteins) often do not contain clear segmentation boundaries, unlike the existence of tokenizable words in many natural languages. This uncertainty in the segmentation of sequences has made overlapping k-mers one of the most popular representations in machine learning for all areas of bioinformatics research, including proteomics (Grabherr et al. 2011; Asgari and M. R. Mofrad 2015), genomics (Jolma et al. 2013; Alipanahi et al. 2015), epigenomics (Awazu 2016; Giancarlo et al. 2015), and metagenomics (Derrick E Wood and Steven L Salzberg 2014; Asgari, Garakani, et al. 2018). However, it is unrealistic to assume that fixed-length k-mers are units of biological sequences and that more meaningful units need to be introduced. This means that although choosing a fixed k value for sequence k-mers simplifies the problem of segmentation, it is an unrealistic assumption to assume that all important part of the sequences have the same length and we need to relax this assumption.

Although in some sequence-labeling tasks (e.g. secondary structure prediction or binding site prediction) sequences are implicitly divided into variable-length segments as the final output, methods to segment sequences into variable-length meaningful units as inputs of downstream machine learning tasks are needed. In § 3.4 nucleotide pair encoding for phenotype and biomarker detection in 16S rRNA data (Asgari, Münch, et al. 2018), which is extended in this work for protein informatics.

Here, we propose a segmentation approach for dividing protein sequences into frequent variable-length sub-sequences, called peptide-pair encoding (PPE). We took the idea of PPE from byte pair encoding (BPE) algorithm, which is a text compression algorithm introduced in 1994 (Gage 1994) that has been also used for compressed pattern matching in genomics (L. Chen et al. 2004). Recently, BPE became a popular word segmentation method in machine translation in NLP for vocabulary size reduction, which also allows for open-vocabulary neural machine translation (Sennrich et al. 2016). In contrast to the use of BPE in NLP for vocabulary size reduction, we used this idea to increase the size of symbols from 20 amino acids to a large set of variable-length frequent sub-sequences, which are potentially meaningful in bioinformatics tasks. In addition, as a modification to the original algorithm, we propose a probabilistic segmentation in a sampling framework allowing for multiple ways of segmenting a sequence into sub-sequences. In particular, we explore the use of PPE for protein sequence motif discovery as well as training embeddings for protein sequences.

***De novo* motif discovery:** Protein short linear motif (SLiM) sequences are short sub-sequences of usually 3 to 20 amino acids that are presumed to have important biological functions; examples of such patterns are cleavage sites, degradation sites, docking sites, ligand binding sites, etc (Prytuliak 2018; Dinkel et al. 2011). Various computational methods have been proposed for the discovery of protein motifs using protein sequence information.

Motif discovery is distinguished from searching for already known motifs in a set of unseen sequences (e.g., SLiMSearch (Davey et al. 2011)). Motif discovery can be done either in a discriminative or a non-discriminative manner. Most of the existing methods are framed as non-discriminative methods, i.e. finding overrepresented protein sub-sequences in a set of sequences of similar phenotype (positive sequences). Examples of non-discriminative methods are SLiMFinder (Edwards et al. 2007) (regular expression based approach), GLAM2 (Frith et al. 2008) (simulated annealing algorithm for alignments of SLiMs), MEME (Bailey et al. 2009) (mixture model fitting by expectation-maximization), HH-MOTiF (Prytuliak, Volkmer, et al. 2017) (retrieving short stretches of homology by comparison of Hidden Markov Models (HMMs) of closely related orthologs). Since other randomly conserved patterns may also exist in the positive sequences, reducing the false positive rate is a challenge for motif discovery (Bingqiang Liu et al. 2017). In order to address this issue, some studies have proposed discriminative motif discovery, i.e. using negative samples or a background set to increase both the sensitivity and specificity of motif discovery. Some examples of discriminative motif miners are DEME (Redhead and Bailey 2007) (using a Bayesian framework over alignment columns), MotifHound (Kelil et al. 2014) (hypergeometric test on certain regular expressions in the input data), DLocalMotif (Mehdi et al. 2013) (combining motif over-representation, entropy and spatial confinement in motif scoring).

Motif databases: General-purpose or specialized datasets are dedicated to maintaining a set of experimentally verified motifs from various resources (e.g., gene ontology). ELM (Dinkel et al. 2011) as a general-purpose dataset of SLiM, and NLSdb (Bernhofer et al. 2017) as a specialized database for nuclear-specific motifs is instances of such efforts. Evaluation of mined motifs can be also subjective. Since the extracted motifs do not always exactly match the experimental motifs, residue-level or site-level evaluations have been proposed (Prytuliak, Volkmer, et al. 2017; Prytuliak, Pfeiffer, et al. 2018). Despite great efforts in this area, computational motif mining has remained a challenging task and the state-of-the-art *de novo* approaches have reported relatively low precision and recall scores, especially at the residue level (Prytuliak, Volkmer, et al. 2017).

Protein embedding: Word embedding has been one of the revolutionary concepts in NLP over the recent years and has been shown to be one of the most effective representations in NLP (Collobert, Weston, et al. 2011; Tang et al. 2014; Levy and Goldberg 2014). In particular, Skip-gram neural networks combined with negative sampling (Mikolov, Sutskever, K. Chen, et al. 2013) has resulted in the state-of-the-art performance in a broad range of NLP tasks (Levy and Goldberg 2014). Recently, we introduced k-mer-based embedding of biological sequences using Skip-gram neural network and negative sampling (Asgari and M. R. Mofrad 2015), which became popular for protein feature extraction and has been extended for various classifications of biological sequences (Asgari and M. R. K. Mofrad 2016; Islam et al. 2017; Min et al. 2017; S. Kim et al. 2018; Jaeger et al. 2018; Du et al. 2018; Hamid and Friedberg 2018).

In this work, inspired by unsupervised word segmentation in NLP, we propose a general-purpose segmentation of protein sequences in frequent variable-length sub-sequences called PPE, as a new representation for machine learning tasks. This segmentation is trained once over large protein sequences (Swiss-Prot) and then is applied to a given set of sequences. In this section, we use this representation for developing a protein motif discovery framework as well as protein sequence embedding.

(i) DiMotif: We suggest a discriminative and alignment-free approach for motif discovery that is capable of finding co-occurred motifs. We do not use sequence alignment; instead, we propose the use of general-purpose segmentation of positive and negative input sequences into PPE sequence segments. Subsequently, we use statistical tests to identify the significant discriminative features associated with the positive class, which are our ultimate output motifs. Being alignment-free makes DiMotif, in particular, a favorable choice for the settings where the positive sequences are not necessarily homologous sequences. In the end, we create sets of multi-part motifs using information theoretic measures on the occurrence patterns of motifs on the positive set. We evaluate DiMotif in the detection of the motifs related to 20 types of experimentally verified motifs and also for searching experimentally verified nuclear localization signal (NLS) motifs. The DiMotif achieved a high recall and a reasonable F1 in comparison with the competitive approaches. In addition, we evaluate a shortlist of extracted motifs on the classification of reserved sequences of the same phenotype for integrins, integrin-binding proteins, and biofilm formation proteins, where the phenotype has been detected with a high F1 score. However, a detailed analysis of the motifs and their biophysical properties are beyond the scope of this study, as the main focus is on introducing the method.

(ii) ProtVecX: We extend our previously proposed protein vector embedding (Protvec) (Asgari and M. R. Mofrad 2015) trained on k-mer segments of the protein sequences to a method of training them on variable-length segments of protein sequences, called ProtVecX. We evaluate our embedding via three protein classification tasks: (i) toxin prediction (binary classification), (ii) subcellular location prediction (four-way classification), and (iii) prediction of enzyme proteins versus non-enzymes (binary classification). We show that concatenation of the raw k-mer distributions with the embedding representations can improve the sequence classification performance over the use of either of k-mers only or embeddings only. In addition, combining of ProtVecX with k-mer occurrence can marginally outperform the use of our originally proposed ProtVec embedding together with k-mer occurrences in toxin and enzyme prediction tasks.

Material and Methods

Datasets

Motif discovery datasets

ELM dataset: The eukaryotic linear motif dataset (ELM) is a commonly used resource of experimentally verified SLiMs. The ELM dataset is usually served as the gold standard for the evaluations of *de novo* motif discovery approaches. Due to the long run-time of DLocalMotif (one of the competitive methods) on large datasets, we perform the evaluation on a subset of ELM. In order to cover a variety of settings, we perform the evaluation on 20 motif discovery problems of 5 different motif types: (i) targeting site (TRG), (ii) post-translational modification sites (MOD), (iii) ligand binding sites (LIG), (iv) docking sites (DOC), and (v) degradation sites (DEG). From each category we randomly select 4 sub-types, which are as distinctive as possible (based on the text similarities in their title). Since the ELM dataset only provides a few accession IDs for each motif sub-types, to obtain more data for the domain-specific segmentation, we expand the positive set using NCBI BLAST. Since our method (DiMotif) and DLocalMotif are both instances of discriminative motif discovery methods, we have to create a background/negative as well. For this purpose, we randomly select sequences from UniRef50 ending up with a dataset 10 times larger than the whole ELM sequence dataset. Then for each motif type, we calculate the sequence 3-mer representation distance between those randomly selected sequences and the positive set. Subsequently, we sample from the randomly selected sequences based on the cosine distance distribution (considering the same size for the positive and the negative classes). This way the farther sequences from the positive set are more likely to be selected in the background or negative set.

Integrin-binding proteins: We extract two positive and negative lists for integrin-binding proteins using the gene ontology (GO) annotation in the UniProt database (U. Consortium 2016). For the positive class, we select all proteins annotated with the GO term GO:0005178 (integrin-binding). Removing all redundant sequences results in 2966 protein sequences. We then use 10% of sequences as a reserved set for evaluation and 90% for motif discovery and training purposes. For the negative class, we select a list of proteins sequences which are annotated with the GO term GO:0005515 (protein binding), but they are annotated neither as integrin-binding proteins (GO:0005178) nor the integrin complex (GO:0008305). Since the resulting set are still large, we limit the selection to reviewed Swiss-Prot sequences and filtered redundant sequences, resulting in 20,117 protein sequences, where 25% of these sequences (5029 sequences) are considered as the negative instances for training and validation, and 297 randomly selected instances (equal to 10% of the positive reserved set) as the negative instances for the negative part of the reserved set.

Integrin proteins: We extract a list of integrin proteins from the UniProt database by selecting entries annotated with the GO term GO:8305 (integrin complex) that also have integrin as part of their entry name. Removing the redundant sequences results in 112 positive sequences. For the negative sequences, we select sequences which are annotated for transmembrane signaling receptor activity (to be similar to integrins) (GO:4888) but which are neither the integrin complex (GO:8305) nor integrin-binding proteins (GO:0005178). Selection of reviewed Swiss-Prot sequences and removal of redundant proteins results in 1155 negative samples. We use 10% of both the positive and negative sequences as the reserved set for evaluation and 90% for motif discovery and training purposes.

Biofilm formation: Similar to integrin-binding proteins, positive and negative lists for biofilm formation are extracted via their GO annotation in UniProt (U. Consortium 2016). For the positive class, we select all proteins annotated with the GO term GO:0042710] (biofilm formation). Removing all redundant sequences results in 1450 protein sequences. For the negative class, we select a list of protein sequences annotated within the parent node of biofilm formation in the GO database that is classified as being for a multi-organism cellular process (GO:44764) but not biofilm formation. Since the number of resulting sequences is large, we limit the selection to reviewed Swiss-Prot sequences and filter the redundant sequences, resulting in 1626 protein sequences. Again, we use 10% of both the positive and negative sequences as a reserved set for evaluation and 90% for motif discovery and training purposes.

Nuclear localization signals: We use the NLSdb dataset containing nuclear export signals and NLS along with experimentally annotated nuclear and non-nuclear proteins (Bernhofer et al. 2017). By using NLSdb annotations from nuclear proteins, we extract a list of proteins experimentally verified to have NLS, ending up with a list of 416 protein sequences. For the negative class, we use the protein sequences in NLSdb annotated as being non-nuclear proteins. NLSdb also contains a list of 3254 experimentally verified motifs, which we use for evaluation purposes.

Protein classification datasets

Sub-cellular location of eukaryotic proteins: The first dataset we use for protein classification is the TargetP 4-classes dataset of sub-cellular locations. The 4 classes in this dataset are (i) 371 mitochondrial proteins, (ii) 715 pathway or signal peptides, (iii) 1214 nuclear proteins, and (iv) 438 cytosolic protein sequences (Emanuelsson et al. 2007), where the redundant proteins are removed.

Toxin prediction: The second dataset we use is the toxin dataset provided by ToxClassifier (Gacesa et al. 2016). The positive set contains 8093 protein sequences annotated in Tox-Prot as being animal toxins and venoms (Jungo and Bairoch 2005). For the negative class, we choose the 'Hard' setting of ToxClassifier (Gacesa et al. 2016), where the negative instances are 7043 protein sequences in UniProt which are not annotated in Tox-Prot but are similar to Tox-Prot sequences to some extent.

Enzyme detection: On the third we use an enzyme classification dataset. We download two lists of enzyme and non-enzyme proteins (22,168 protein sequences per class) provided by the 'NEW' dataset of Deepre (Y. Li et al. 2017).

Peptide-pair encoding

PPE training

The input to the PPE algorithm is a set of sequences and the output would be segmented sequences and segmentation operations, an ordered list of amino acid merging operations to be applied for segmenting new sequences. At the beginning of the algorithm, we treat each sequence as a list of amino acids. As detailed in Algorithm 1, we then search for the most frequently occurring pair of adjacent amino acids in all input sequences. In the next step, the select pairs of amino acids are replaced by the merged version of the selected pair as a new symbol (a short peptide). This process is continued until we could not find a frequent pattern or we reach a certain vocabulary size (Algorithm 1).

In order to train a general-purpose segmentation of protein sequences, we train the segmentation over the most recent version of the Swiss-Prot database (Boutet et al. 2016), which contained 557,012 protein sequences (step (i) in Figure 2.5). We continue the merging steps for T iterations of Algorithm 1, which ensures that we capture the motifs present with a minimum frequency f in all Swiss-Prot sequences (we set the threshold to a minimum of $f = 10$ times, resulting in $T \approx 1$ million iterations). Subsequently, the merging operations can be applied to any given protein sequences as a general-purpose splitter. Although there exist larger sequence datasets than Swiss-Prot (e.g., UniProt and RefSeq), we decided to use the Swiss-Prot database, due to the high quality, and as computational requirements were less demanding. Principally, PPE segmentations can be generated from any database of interested, given enough time and computational resources.

Monte Carlo PPE segmentation

The PPE algorithm for a given vocabulary size (which is analogous to the number of merging steps in the training) divides a protein sequence into a unique sequence of sub-sequences. Further merging steps result in enlargement of sub-sequences, which results in having fewer sub-sequences. Such variations can be viewed as multiple valid schemes of sequence segmentation. For certain tasks, it might be useful to consider a protein sequence as a chain of residues and, in some cases, as a chain of large protein domains. Thus, sticking to a single segmentation scheme will result in ignoring important information for the task of interest. In order to address this issue, we propose a sampling framework for estimating the segmentation of a sequence in a probabilistic manner. We sample from the space of possible segmentations for both motif discovery and embedding creation.

Different segmentation schemes for a sequence can be obtained by a varying number of merging steps (N) in the PPE algorithm. However, since the algorithm is trained over a

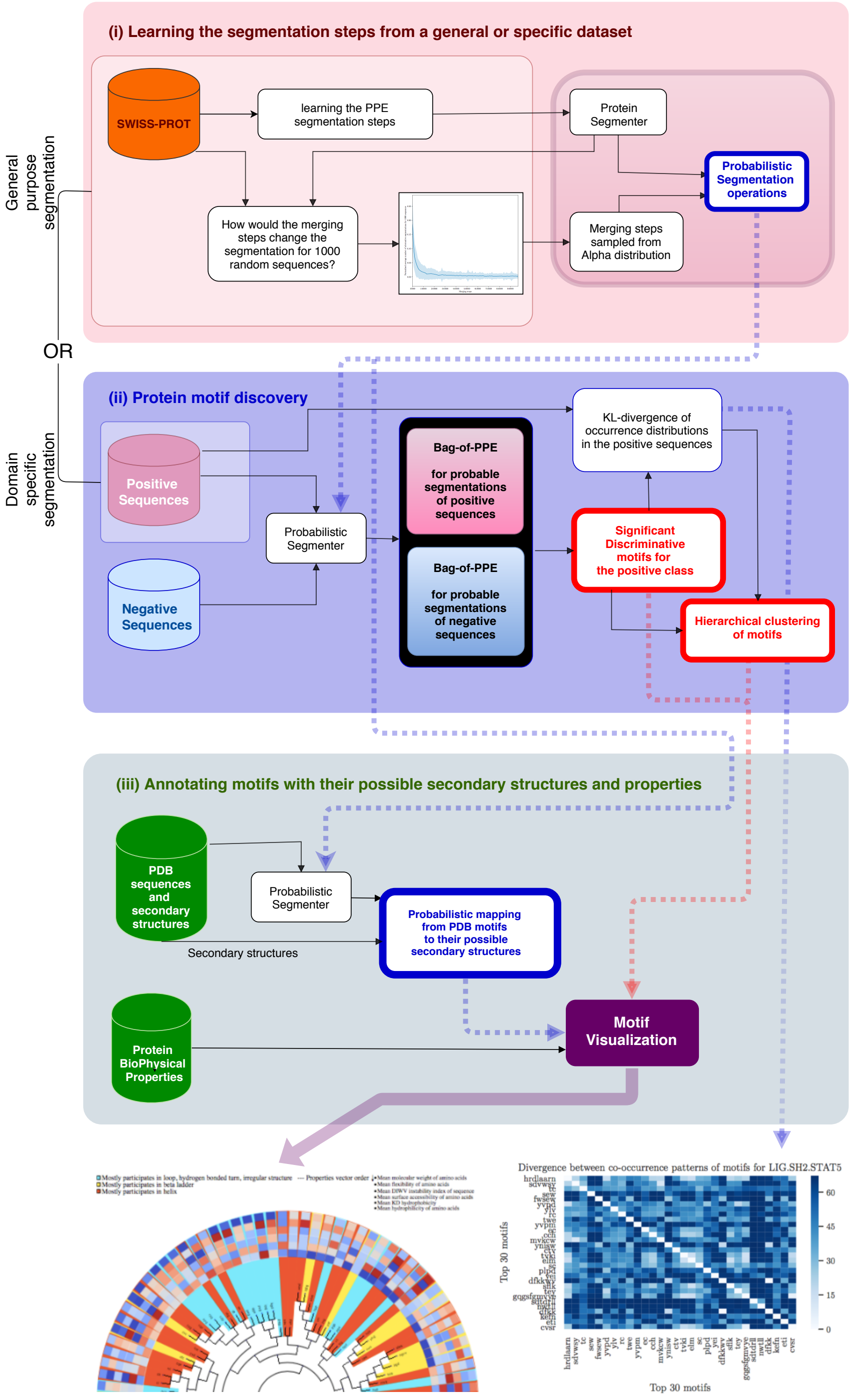

(i) Learning the segmentation steps from a general or specific dataset
SWISS-PROT
learning the PPE segmentation steps
Protein Segmenter
Probabilistic Segmentation operations
How would the merging steps change the segmentation for 1000 random sequences?
Merging steps sampled from Alpha distribution
General purpose segmentation
OR
Domain specific segmentation
(ii) Protein motif discovery
Positive Sequences
Negative Sequences
Probabilistic Segmenter
Bag-of-PPE for probable segmentations of positive sequences
Bag-of-PPE for probable segmentations of negative sequences
KL-divergence of occurrence distributions in the positive sequences
Significant Discriminative motifs for the positive class
Hierarchical clustering of motifs
(iii) Annotating motifs with their possible secondary structures and properties
PDB sequences and secondary structures
Probabilistic Segmenter
Probabilistic mapping from PDB motifs to their possible secondary structures
Secondary structures
Protein BioPhysical Properties
Motif Visualization
Mostly participates in loop, hydrogen bonded turn, irregular structure
Mostly participates in beta ladder
Mostly participates in helix
Properties vector order
Mean molecular weight of amino acids
Mean flexibility of amino acids
Mean DIWV instability index of sequence
Mean surface accessibility of amino acids
Mean KD hydrophobicity
Mean hydrophilicity of amino acids
Divergence between co-occurrence patterns of motifs for LIG.SH2.STAT5
Top 30 motifs
Top 30 motifs
hrdlaarn
sdvwsy
tc
sew
fwsew
yvpd
ylv
rc
twe
yvpm
ec
cch
mvkcw
ynisw
ctv
tvki
elm
sc
plpd
yei
dfkkwv
sflk
tey
gqgsfgmvye
sdtdrll
nwtll
dfkk
kefn
eti
cvsr
60
45
30
15
0

Figure 2.5. The main steps of DiMotif computational workflow: (i) The PPE segmentation steps can be learned from Swiss-Prot or a domain-specific set of sequences. (ii) These operations are then applied to positive and negative sequences and segment them into smaller sub-sequences. This means that all part of sequences are used till this part. A two-sided and FDR corrected χ^2 test is then applied to find the sub-sequences (potential motifs) which are significantly related to the positive class, with a threshold p-value < 0.05. We rank the motifs based on their significance and retrieve the top-k (in the evaluation on the ELM dataset $k_{max} = 30$). (iii) The motifs, their structural and biophysical properties, and their co-occurrence information will be used for visualization.

```
Data: $Seqs$ = Set of Swiss-Prot protein sequences, $f$ = minimum number of sequences
      containing the newly emerged symbol
Result: $S$ = Divided sequences into variable sub-sequences, $Merge\_opt$ = merging
        operations
$Sym = \{A, H, K, T, E, C, V, N, W, Y, F, Q, G, P, D, L, S, R, M, I\}$;
$S$ = list of $Seqs$, where each sequence is a list of symbols $\in Sym$;
$Merge\_opt = stack()$;
$SymbFreq$ = mapping symbol pairs in $S$ to their frequencies;
$f_{current}$ = max frequency in $SymbFreq$;
while $f < f_{current}$ do
    sym1, sym2 = argmax ($SymbFreq$);
    $S$ = merge all consecutive sym1 & sym2 into $< sym1, sym2 >$ in $S$;
    $Sym.push(< sym1, sym2 >)$;
    $Merge\_opt.push(sym1, sym2)$;
    update($SymbFreq$);
    $current_f$ = max frequency in $SymbFreq$;
end
```

Algorithm 1: Adapted Byte-pair algorithm (BPE) for segmentation of protein sequences

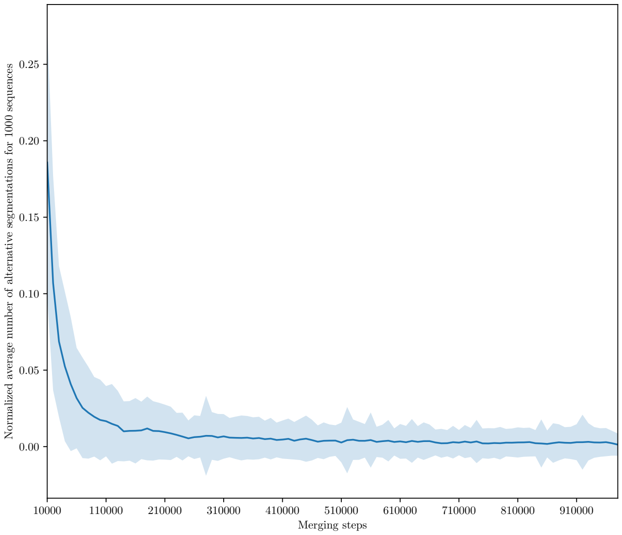


Figure 2.6. Average number of segmentation alternation per merging steps for 1000 Swiss-prot sequences.

large number of sequences, a single merging step will not necessarily affect all sequences, and as we go further with merging steps, fewer sequences are affected by the newly introduced symbol. We estimate the probability density function of possible segmentation schemes with respect to N by averaging the segmentation alternatives over 1000 random sequences in Swiss-Prot for $N \in [10000, 1000000]$, with a step size of 10000. For each N, we count the average number of introduced symbols relative to the previous step; the average is shown in Figure 2.6. We use this distribution to draw samples from the vocabulary sizes that affected more sequences (i.e. those introducing more alternative segmentation schemes). To estimate this empirical distribution with a theoretical distribution, we fit a variety of distributions (Gaussian; Laplacian; and *Alpha*, *Beta*, and *Gamma* distributions) using maximum likelihood and found the *Alpha* that fitted the empirical distribution the best. Subsequently, we use the fitted Alpha distribution to draw segmentation samples from a sequence in a Monte Carlo scheme. Consider $\Phi_{i,j}$ as the $l1$ normalized "bag-of-word" representations in sequence i, using the j^{th} sample from the fitted *Alpha* distribution. Thus, if we have M samples, the "bag-of-sub-sequences" representation of the sequence i can be estimated as $\Phi_i = \frac{1}{M}\sum_j^M \Phi_{i,j}$.

For detecting sequence motifs, we represented each sequence as an average over the count distribution of M samples of segmentation (M=100) drawn from the *Alpha* distribution. The alternative is to use only the vocabulary size (e.g., the median of *Alpha*), referred to as the non-probabilistic segmentation in this section.

DiMotif protein sequence motif discovery

Our proposed method for motif detection, called DiMotif, finds motifs in a discriminative setting over PPE features (step (ii) in Figure 2.5). We segment all the sequences in the datasets (ignoring their labels or their membership of the train or the test set) with the learned PPE segmentation steps from Swiss-Prot (§2.3) (general-purpose segmentation) or with the learned PPE segmentation steps from a set of positive sequences (domain-specific segmentation). After segmentation, each sequence is represented as bag-of-PPE units. We use a two-sided and FDR corrected χ^2 test to identify significant discriminative features between the positive and the negative (or background) sets. We discard insignificant motifs using a threshold for the p-value of < 0.05. Since we are looking for sequence motifs related to the positive class, we exclude motifs related to the negative class.

Evaluation on ELM dataset: We compare the DiMotif performance with two recent motif discovery tools: (i) HH-Motif (Prytuliak, Volkmer, et al. 2017) as an instance of non-discriminative methods and (ii) DLocalMotif (Mehdi et al. 2013) as an instance of discriminative approaches. We evaluate the performances over the 20 problem settings related to 5 types of motifs in the ELM database. We measure precision and recall of the above methods for detection of the experimentally verified motifs (as true positive). From each method, we use the maximum top 30 retrieved motifs ranked based on their scores. Since finding exact matches are very unlikely and the motifs are only correct to some extent. We report precision and recall for different thresholds on motif sequence matching (50% and

70%). Then we calculate the average precision, recall, and F1 on these different settings. In order to investigate the performance of general-purpose versus domain-specific segmentations in DiMotif, once we used Swiss-Prot segmentation and once we learned the segmentation steps from the set of positive sequences and for both used the probabilistic segmentation schemes.

Classification-based evaluation of integrins, integrin-binding proteins, and biofilm formation motifs: In order to evaluate the obtained motifs, we train linear support vector machine classifiers over the training instances but only use motifs related to the positive class among the top 1000 motifs as well as a short list of features. Next, we test the predictive model on a reserved test set. Since the training and testing sets are disjoint, the classification results are indications of motif discovery quality. We use both probabilistic and non-probabilistic segmentation methods to obtain PPE representations of the sequences. We report the precision, recall, and F1 of each classifier's performance. The average sequence similarities for the top hits between positive samples in the test set and the train set for integrins, integrin binding, and biofilm formation were 35.50±14.41, 40.47±18.15, and 40.13±8.76 respectively. In addition, the average sequence similarities for the best hits for integrins, integrin binding, and biofilm formation were 83.96±11.96, 91.64±11.37, and 71.75±15.79 respectively.

NLS motifs search: In the case of NLS motifs, we use the list of 3254 experimentally or manually verified motifs from NLSdb. Thus, in order to evaluate our extracted motifs, we directly compare our motifs with those found in earlier verification. As we cannot evaluate any true positive other than NLSdb this task can be considered as a motif search task. Since for long motifs, finding exact matches is challenging, we report three metrics, the number of motifs with at least three consecutive amino acid overlaps, the number of sequences in the baseline that had a hit with more than 70% overlap (A to B and B to A), and finally the number of exact matches. In addition to Swiss-Prot-based segmentation, in order to see the effect of a specialized segmentation, we also train PPE segmentation over a set of 8421 nuclear protein sequences provided by NLSdb (Bernhofer et al. 2017) and perform the same evaluation.

Kulback-Leibler divergence to find multi-part motifs

As discussed in§ 2.3, protein motifs can be multi-part patterns, which is ignored by many motif-finding methods. In order to connect the separated parts, we propose to calculate the symmetric Kullback–Leibler (KL) divergence (Kullback and Leibler 1951) between motifs based on their co-occurrences in the positive sequences as follows:

$$D_{\mathrm{KL}_{\mathrm{sym}}}(M_p \| M_q) = \sum_{i}^{N} M_p(i) \log \frac{M_p(i)}{M_q(i)} + M_q(i) \log \frac{M_q(i)}{M_p(i)}, \tag{2.4}$$

where M_p and M_q are, respectively, the normalized occurrence distributions of motif p and motif q across all positive samples and N is the number of positive sequences. Next, we use the condition of ($D_{\mathrm{KL}_{\mathrm{sym}}} = 0$) to find co-occurring motifs splitting the motifs into equivalence classes. Each equivalent class indicates a multi-part or a single-part motif. Since we considered a “bag of motifs" assumption, the parts of multi-part motifs are allowed to be far from each other in the primary sequence.

Secondary structure assignment

Using the trained segmentation over the Swiss-Prot sequences, we segment all 385,937 protein sequences in the current version of the PDB (Rose et al. 2016), where their secondary structure was provided. By segmenting all secondary structures at the same positions as the corresponding sequences, we obtain a mapping from each sequence segment to all its possible secondary structures in the PDB. We use this information in coloring in the visualization of motifs (see the visualizations in §2.3).

Motif visualization: For visualization purposes DiMotif clusters motifs based on their co-occurrences in the positive class by using hierarchical clustering over the pairwise symmetric KL divergence. The motifs are then colored based on the most frequent secondary structure they assume in the sequences in the Protein Data Bank (PDB) (step (iii) in Figure 2.5). For each motif, it visualizes their mean molecular weight, mean flexibility (Vihinen et al. 1994), mean instability (Guruprasad et al. 1990), mean surface accessibility (Emini et al. 1985), mean kd hydrophobicity (Kyte and Doolittle 1982), and mean hydrophilicity (Hopp and Woods 1981) with standardized scores (zero mean and unit variance), where the dark blue is the lowest and the dark red is the highest possible value (see the visualizations in §2.3).

ProtVecX: Extended variable-length protein vector embeddings

We trained the embedding on segmented sequences obtained from Monte Carlo sampling segmentation on the most recent version of the Swiss-Prot database (Boutet et al. 2016), which contains 557,012 protein sequences. Since this embedding is the extended version of ProtVec (a brief introduction to the ProtVec is provided in the Supp. §1), we call it ProtVecX. As explained in §2.3 we segment each sequence with the vocabulary size samples drawn from an *Alpha* distribution. This ensures that we consider multiple ways of segmenting sequences during the embedding training. Subsequently, we train a Skip-gram neural network for embedding on the segmented sequences (Mikolov, Sutskever, K. Chen, et al. 2013). The Skip-gram neural network is analogous to language modeling, which predicts the surroundings (context) for a given textual unit (shown in Figure 2.7). The Skip-gram's objective is to maximize the following log-likelihood:

$$\sum_{t=1}^{M} \sum_{c\in[t-N,t+N]} \log p(w_c \mid w_t), \tag{2.5}$$

where N is the surrounding window size around word w_t, c is the context indices around index t, and M is the corpus size in terms of the number of available words and context pairs. We parameterize this probability of observing a context word w_c given w_t by using word embedding:

$$p(w_c \mid w_t; \theta) = \frac{e^{v_c \cdot v_t}}{\sum_{c' \in \mathcal{C}} e^{v_{c'} \cdot v_t}}, \tag{2.6}$$

where $\mathcal{C}$ denotes all existing contexts in the training data. However, iterating over all existing contexts is computationally expensive. This issue can be efficiently addressed by using negative sampling. In a negative sampling framework, we can rewrite Equation 2.5 as follows:

$$\sum_{t=1}^{T} \left[\sum_{c \in [t-N, t+N]} \log\left(1 + e^{-s(w_t,\ w_c)}\right) + \sum_{w_r \in \mathcal{N}_{t,c}} \log\left(1 + e^{s(w_t,\ w_r)}\right) \right], \tag{2.7}$$

where $\mathcal{N}_{t,c}$ denotes a set of randomly selected negative examples sampled from the vocabulary collection as non-contexts of w_t and $s(w_t,\ w_c) = {v_t}^\top \cdot v_c$ (parameterization with the word vector v_t and the context vector v_c). For training embeddings on PPE units, we used the sub-word level Skip-gram, known as fasttext (Bojanowski et al. 2017). Fasttext embedding improves the word representations by taking character k-mers of the sub-words into consideration in calculating the embedding of a given word. For instance, if we take the PPE unit *fggagvg* and $k = 3$ as an example, it will be represented by the following character 3-mers and the whole word, where '<' and '>' denote the start and the end of a PPE unit:

$$\mathcal{S}_{fggagvg}=\{\text{`<fg'}, \text{`fgg'}, \text{`gga'}, \text{`gag'}, \text{`agv'}, \text{`gvg'}, \text{`vg>'}, \text{`<fggagvg>'}\}$$

In the fasttext model, the scoring function will be based on the vector representation of k-mers ($2 \leq k \leq 6$) that exist in textual units (PPE units in this case), $s(w_t, w_c) = \sum_{x \in \mathcal{S}_{w_t}} v_x^\top v_c$.

We used a vector dimension of 500 for the embedding (v_t's) and a window size of 20 (the vector size and the window size have been selected based on a systematic exploration of parameters in protein classification tasks). A k-mer-based ProtVec of the same vector size and the same window size trained on Swiss-Prot is used for comparison.

Embedding-based classification

For the classification, we use a Multi-Layer-Perceptrons (MLP) neural network architecture with five hidden layers using Rectified Linear Unit (ReLU) as the nonlinear activation function. We use the softmax activation function at the last layer to produce the probability vector that could be regarded as representing posterior probabilities. To avoid overfitting, we perform early stopping and also use dropout at hidden layers. As baseline representations, we use k-mers, ProtVec (Asgari and M. R. Mofrad 2015), ProtVecX, and their combinations. For both ProtVec and ProtVecX, the embedding of a sequence is calculated as the summation of

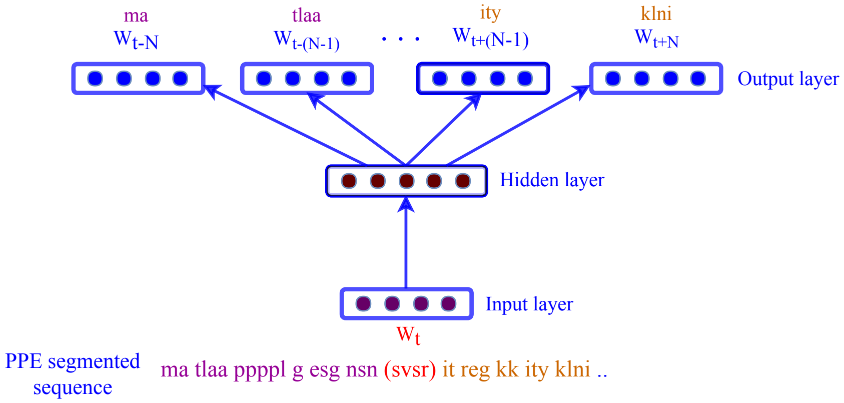


Figure 2.7. Skip-gram neural network for training language model-based embedding. In this framework the inputs are the segmented sequences and the network is trained to predict the surroundings PPE units.

its k-mers or PPE unit vectors. We evaluate these representation in three protein classification tasks: (i) toxin prediction (binary classification) with the 'Hard' setting in the ToxClassifier database (Gacesa et al. 2016), (ii) subcellular location prediction (four-way classification) using the dataset provided by TargetP (Emanuelsson et al. 2007), and (iii) prediction of enzyme proteins versus non-enzymes (binary classification) using the NEW dataset (Y. Li et al. 2017). We report macro- precision, recall, and F-1 score. Macro averaging computes the metrics for each class separately and then simply averages over classes. This metric gives equal importance to all categories. In particular, we are interested in macro-F1, which makes a trade-off between precision and recall in addition to treating all classes equally.

Results

Sequence motifs and evaluation results

Detection of experimentally verified motifs in the ELM dataset A comparison of DiMotif with two existing motif discovery tools, HH-Motif (non-discriminative) and DLocal-Motif (discriminative), is provided in Table 2.5. Since the discovered motifs only partially match the experimentally verified motifs, we measured precision, recall, and F1 scores for two minimum ratios of 50% and 70% sequence matching between the computationally discovered and the experimentally verified motifs (two sets of rows in Table 2.5 for two minimum sequence matching ratios). DiMotif was used with two different schemes of segmentation: (i) general-purpose segmentation (based-on Swiss-Prot) and (ii) domain-specific segmentation (learned over the sequences in the positive class). Overall, HH-Motif achieved the best F1 scores of 0.39 for 50% sequence matching and F1 of 0.24 for 70% sequence matching. The domain-specific DiMotif obtained F1 of 0.30 for 50% sequence matching and F1 of 0.07 for 70% sequence matching. The general-purpose DiMotif obtained F1 of 0.24 for 50% sequence matching and F1 of 0.05 for 70% sequence matching, while DLocalMotif obtained F1 of 0.16

		HH-MOTIF			**DiMotif** (Domain-specific)			**DiMotif** (SWISS-Prot-based)			**DLocalMotif**		
	Motif discovery setting	Precision	Recall	F1 score	Precision	Recall	F1 score	Precision	Recall	F1 score	Precision	Recall	F1 score
Matching condition: **50%** overlap between motifs	DEG APCC KENBOX 2	0.80	1.00	0.89	0.07	0.94	0.13	0.26	1.00	0.41	0.23	0.81	0.36
	DEG CRL4 CDT2 1	0.60	1.00	0.75	0.23	0.83	0.36	0.00	0.00	0.00	0.00	0.00	0.00
	DEG Kelch KLHL3 1	0.00	0.00	0.00	0.03	1.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00
	DEG SPOP SBC 1	0.00	0.00	0.00	0.03	0.14	0.05	0.20	0.71	0.31	0.07	0.86	0.13
	DOC AGCK PIF 3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	DOC MAPK 2	0.59	1.00	0.74	0.17	1.00	0.29	0.33	1.00	0.50	0.00	0.00	0.00
	DOC PIKK 1	0.33	0.50	0.40	0.17	1.00	0.29	0.10	0.75	0.18	0.00	0.00	0.00
	DOC USP7 MATH 1	0.00	0.00	0.00	0.30	0.60	0.40	0.20	0.60	0.30	0.00	0.00	0.00
	LIG 14-3-3 2	0.08	0.28	0.13	0.30	1.00	0.46	0.17	0.43	0.24	0.10	0.43	0.16
	LIG Mtr4 Air2 1	0.00	0.00	0.00	0.13	1.00	0.23	0.00	0.00	0.00	0.00	0.00	0.00
	LIG SH2 STAT5	0.29	0.38	0.33	0.47	1.00	0.64	0.30	1.00	0.46	0.00	0.00	0.00
	LIG TYR ITIM	0.25	0.17	0.20	0.27	0.83	0.41	0.17	0.50	0.25	0.27	0.67	0.38
	MOD CDK 1	0.30	0.70	0.42	0.20	0.90	0.33	0.30	0.90	0.45	0.53	0.80	0.64
	MOD NEK2 1	0.33	0.66	0.44	0.17	1.00	0.29	0.04	0.33	0.07	0.00	0.00	0.00
	MOD SPalmitoyl 4	1.00	1.00	1.00	0.17	1.00	0.29	0.14	1.00	0.25	0.53	1.00	0.69
	MOD SUMO rev 2	0.17	0.05	0.08	0.37	0.79	0.50	0.37	0.53	0.44	0.30	0.21	0.25
	TRG AP2beta CARGO 1	0.50	1.00	0.67	0.10	0.50	0.17	0.05	0.75	0.09	0.03	0.25	0.05
	TRG ER KDEL 1	0.60	1.00	0.75	0.37	1.00	0.54	0.43	1.00	0.60	0.23	0.20	0.21
	TRG LysEnd APsAcLL 3	1.00	1.00	1.00	0.20	1.00	0.33	0.17	1.00	0.29	0.00	0.00	0.00
	TRG NES CRM1 1	0.08	0.06	0.07	0.17	0.56	0.26	0.00	0.00	0.00	0.00	0.00	0.00
Macro-Averaged Metrics		0.35	0.49	0.39	0.20	0.80	0.30	0.16	0.57	0.24	0.11	0.26	0.16
Matching condition: **70%** overlap between motifs	DEG APCC KENBOX 2	0.52	0.88	0.65	0.03	0.94	0.06	0.10	0.38	0.16	0.03	0.06	0.04
	DEG CRL4 CDT2 1	0.52	1.00	0.68	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	DEG Kelch KLHL3 1	0.00	0.00	0.00	0.03	0.50	0.06	0.00	0.00	0.00	0.00	0.00	0.00
	DEG SPOP SBC 1	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.71	0.27	0.00	0.00	0.00
	DOC AGCK PIF 3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	DOC MAPK 2	0.24	1.00	0.39	0.07	1.00	0.13	0.07	0.00	0.00	0.00	0.00	0.00
	DOC PIKK 1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	DOC USP7 MATH 1	0.00	0.00	0.00	0.03	0.10	0.05	0.13	0.30	0.18	0.00	0.00	0.00
	LIG 14-3-3 2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	LIG Mtr4 Air2 1	0.00	0.00	0.00	0.07	1.00	0.13	0.00	0.00	0.00	0.00	0.00	0.00
	LIG SH2 STAT5	0.00	0.00	0.00	0.13	0.50	0.21	0.13	0.25	0.17	0.00	0.00	0.00
	LIG TYR ITIM	0.00	0.00	0.00	0.03	0.16	0.05	0.00	0.00	0.00	0.13	0.33	0.19
	MOD CDK 1	0.30	0.20	0.24	0.03	0.30	0.05	0.00	0.00	0.00	0.07	0.20	0.10
	MOD NEK2 1	0.00	0.00	0.00	0.03	0.33	0.05	0.00	0.00	0.00	0.00	0.00	0.00
	MOD SPalmitoyl 4	1.00	1.00	1.00	0.30	0.20	0.24	0.00	0.00	0.00	0.13	0.40	0.20
	MOD SUMO rev 2	0.00	0.00	0.00	0.10	0.16	0.12	0.03	0.05	0.04	0.00	0.00	0.00
	TRG AP2beta CARGO 1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	TRG ER KDEL 1	0.60	1.00	0.75	0.13	0.80	0.22	0.13	0.80	0.22	0.23	0.20	0.21
	TRG LysEnd APsAcLL 3	1.00	1.00	1.00	0.03	0.33	0.05	0.00	0.00	0.00	0.00	0.00	0.00
	TRG NES CRM1 1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Macro-Averaged Metrics		0.21	0.30	0.24	0.05	0.32	0.07	0.04	0.12	0.05	0.03	0.06	0.04

Table 2.5. Comparison of DiMotif, HH-Motif, and DLocalMotif (four sets of columns) performances in detection of experimentally verified motifs in the ELM dataset. Two versions of DiMotif were used: (i) using Swiss-Prot segmentation and (ii) using a domain specific segmentation. The performances are reported for 50% and 70% ratios of sequence matching (two sets of rows) between the verified and the discovered motifs.

for 50% sequence matching and F1 of 0.04 for 70% sequence matching. The domain-specific DiMotif achieved the maximum recall of 0.80 for 50% sequence matching and F1 of 0.32 for 70% sequence matching. Having high recall suggests that the DiMotif can be used for short-list creation for further experimental investigations on motifs.

Table 2.6. Evaluation of protein sequence motifs mined via PPE motif discovery for classification of integrin-binding proteins and biofilm formation-associated proteins. Support Vector Machine classifiers are tuned and evaluated in a stratified 10-fold cross-validation setting and then tested on a separate reserved dataset.

Dataset	Probabilistic Segmentation	Representation	10-fold cross-validation			Performance on the test set		
			Precision	Recall	F1	Precision	Recall	F1
Integrin-Binding	True	top 1000 (998 positive)	0.88	0.85	0.87	0.91	0.85	0.88
		top-100 (100 positive)	0.73	0.66	0.69	0.84	0.67	0.75
	False	top 1000 (982 positive)	0.91	0.87	0.89	0.93	0.86	**0.89**
		top-100 (100 positive)	0.73	0.68	0.70	0.84	0.67	0.75
Integrins	True	top 1000 (1000 positive)	0.94	0.76	0.84	1	0.75	0.86
		top-100 (100 positive)	0.91	0.82	0.86	0.83	0.83	**0.83**
	False	top 1000 (996 positive)	0.96	0.82	0.89	1	0.83	0.91
		top-100 (100 positive)	0.88	0.83	0.86	0.9	0.75	0.82
Biofilm formation	True	top-1000 (103 positive)	0.89	0.67	0.76	0.82	0.56	0.72
		top-500 (48 positive)	0.81	0.71	0.76	0.76	0.71	**0.73**
	False	top 1000 (53 positive)	0.79	0.67	0.73	0.74	0.66	0.70
		top-500 (26 positive)	0.78	0.67	0.72	0.73	0.65	0.69

Classification-based evaluation of integrins, integrin-binding, and biofilm formation motifs: The performances of machine learning classifiers in phenotype prediction using the extracted motifs as features are provided in Table 2.6 evaluated in both 10-fold cross-validation scheme, as well as in classifying unseen reserved sequences. Both probabilistic and non-probabilistic segmentation methods have been used to obtain PPE motifs. However, from the top extracted motifs only motifs associated with the positive class are used as features (representation column). For each classification setting, we report precision, recall, and F1 scores. The trained classifiers over the extracted motifs associated with the positive class could reliably predict the reserved integrins, integrin-binding proteins, and biofilm formation proteins with F1 scores of 0.89, 0.89, and 0.75 respectively. As described in §2.3 the sequences with certain degrees of redundancy were already removed and the training data and the reserved sets do not overlap. Thus, being able to predict the phenotype over the reserved sets with high F1 scores shows the quality of motifs extracted by DiMotif. This confirms that the extracted motifs are specific and inclusive enough to detect the phenotype of interest among an unseen set of sequences.

For integrin and biofilm formation, the probabilistic segmentation helps in predictions of the reserved dataset. This suggests that multiple views of segmenting sequences allows the statistical feature selection model to be more inclusive in observing possible motifs. Picking a smaller fraction of positive class motifs still resulted in a high F1 for the test sets. For biofilm formation, the probabilistic segmentation improved the classification F1 score from 0.72 to 0.73 when only 48 motifs were used, where single segmentation even using more features obtained an F1 score of 0.70 (Table 2.6). This classification result suggests that the only 48 motifs mined from the training set are enough to detect bioform formation proteins in the test set. Thus, such a combination can be a good representative of biofilm formation motifs.

Literature-based evaluation of NLS motifs: Since NLSdb provided us with an extensive

Table 2.7. Evaluation of the significant nuclear localization signal (NLS) patterns against 3254 experimentally identified motifs. The results are provided for both general purpose and domain-specific segmentation of sequences.

PPE training dataset	Probabilistic Segmentation	Medium overlap: Overlapping hits (> 3)	Large overlap: > 70% sequence overlap	Number of exact matches
Swiss-Prot (General purpose)	True	3253	337	37
Swiss-Prot (General purpose)	False	3162	107	15
Nuclear (Domain-specific)	True	3253	381	42
Nuclear (Domain-specific)	False	3198	137	21

list of experimentally verified NLS motifs, we evaluated the extracted motifs by measuring their overlap with NLSdb instead of using a classification-based evaluation. However, as discussed in §2.3 such a comparison can be very challenging. One reason is that different methods and technologies vary in their resolutions in specifying the motif boundaries. In addition, the motifs extracted by the computational methods may also contain some degrees of false negatives and false positives. Thus, instead of reporting exact matches in the experimentally verified set, we report how many of 3254 motifs in NLSdb are verified by our approach using three different degrees of similarity (medium overlap, large overlap, and exact match). The performance of DiMotif for both probabilistic segmentation and non-probabilistic segmentation are provided in Table 2.7. In order to investigate the performance of phenotype-specific versus general purpose segmentation, we also report the results based on segmentation that is learned from nuclear proteins, in addition to Swiss-Prot based segmentation (which is supposed to general purpose). Training the segmentation on nuclear proteins resulted in slightly better, but still competitive to the general-purpose Swiss-Prot segmentation. This result shows that the segmentation steps learned from Swiss-Prot can be considered as a general segmentation, which is important for low resource settings, i.e. the problem setting that the number of positive samples is relatively small. Similar to integrins and biofilm formation related proteins, the probabilistic segmentation has been more successful in detecting experimentally verified NLS motifs as well (Table 2.7).

DiMotif Visualization: The top extracted motifs are visualized using DiMotif software and are provided for interested readers, related to integrin-binding proteins (Figure 2.8), biofilm formation (Figure 2.9), and integrin complexes (Figure 2.10). In these visualizations, motifs are clustered according to their co-occurrences within the positive set, i.e. if two motifs tend to occur together (not necessarily close in the linear chain) in these hierarchical clustering they are in a close proximity. In addition, each motif is colored based on the most frequent secondary structure that this motif can assume in all existing PDB structures (described in §2.3), the blue background shows loop, hydrogen bound or irregular structures, the yellow background shows β-sheet or β-bridge conformations, and red background shows helical structures. Furthermore, to facilitate the interpretation of the found motifs, DiMotif provides a heatmap representation of biophysical properties related to each motif, namely molecular weight, flexibility, instability, surface accessibility, kd hydrophobicity, and mean hydrophilicity with standardized scores (zero mean and unit variance) the dark blue is the lowest and the dark red is the highest possible value. Normalized scores allow for an easier visual comparison. For instance, interestingly in most cases in the trees (Figure 2.8, Figure 2.9, and Figure 2.10), the neighbor motifs (co-occurring motifs) agree in their frequent secondary structures. Furthermore, some neighbor motifs agree in some provided biophysical properties. Such information can assist biologists and biophysicists to make hypotheses about the underlying motifs and mechanisms for further experiments. A detailed serious biophysical investigation of the extracted motifs is beyond the scope of this study. However, as an example, for integrin-binding proteins, the RGD motif, the most well-known integrin-binding motif was among the most significant motifs in our approach (Guan and Hynes 1990; Ruoslahti 1996; Plow, T. A. Haas, et al. 2000). Other known integrin-binding motifs were also among the most significant ones, such as RLD (Ruoslahti 1996), KGD (the binding site for the $\alpha II\beta 3$ integrins (Plow, Pierschbacher, et al. 1985)), GPR (the binding site for $\alpha_x\beta 2$ (Plow, T. A. Haas, et al. 2000)), LDT (the binding site for $\alpha_4\beta 7$ (Plow, T. A. Haas, et al. 2000)), QIDS (the binding site for $\alpha_4\beta 1$ (Plow, T. A. Haas, et al. 2000)), DLLEL (the binding site for $\alpha_v\beta 6$ (Plow, T. A. Haas, et al. 2000)), [tldv,rldvv,gldvs] (similar motifs to LDV, the binding site for $\alpha_4\beta 1$ (Guan and Hynes 1990)), rgds (Kapp et al. 2017), as well as the PEG motif (Ochsenhirt et al. 2006).

Results of protein classification tasks using embedding

Protein classification results for venom toxins, subcellular location, and enzyme predictions using deep MLP neural network on top of different combinations of features are provided in Table 2.8. In all these three tasks, combining the embeddings with raw k-mer distributions improves the classification performances (Table 2.8). This result suggests that k-mers can be more specific than embeddings for protein classification. However, embeddings can provide complementary information to the k-mers and improve the classification performances. Combining 3-mers with either ProtVecX or ProtVec embedding performed very competitively; even for sub-cellular prediction tasks, ProtVec performs slightly better. However, combining 3-mers with ProtVecX resulted in higher F1 scores for enzyme classification and toxin protein prediction. In § ProtVec 2.2 as well as other embedding-based protein classification studies (Hamid and Friedberg 2018), embeddings have been used as the only representation. The presented results in Table 2.8 suggest that (i) k-mer representation is a tough-to-beat baseline in protein classification problems. (ii) The k-mer baseline alone outperforms ProtVec embeddings in many tasks. (iii) The ProtVec and ProtVecX embeddings only have added value when they are combined with the raw k-mer representations.

Conclusions

We proposed a new unsupervised method of feature extraction from protein sequences. Instead of fixed-length k-mers, we segmented sequences into the most frequent variable-length sub-sequences, inspired by BPE, a data compression algorithm. These sub-sequences were then used as features for downstream machine learning tasks. As a modification to the original BPE algorithm, we defined a probabilistic segmentation by sampling from the space of possible

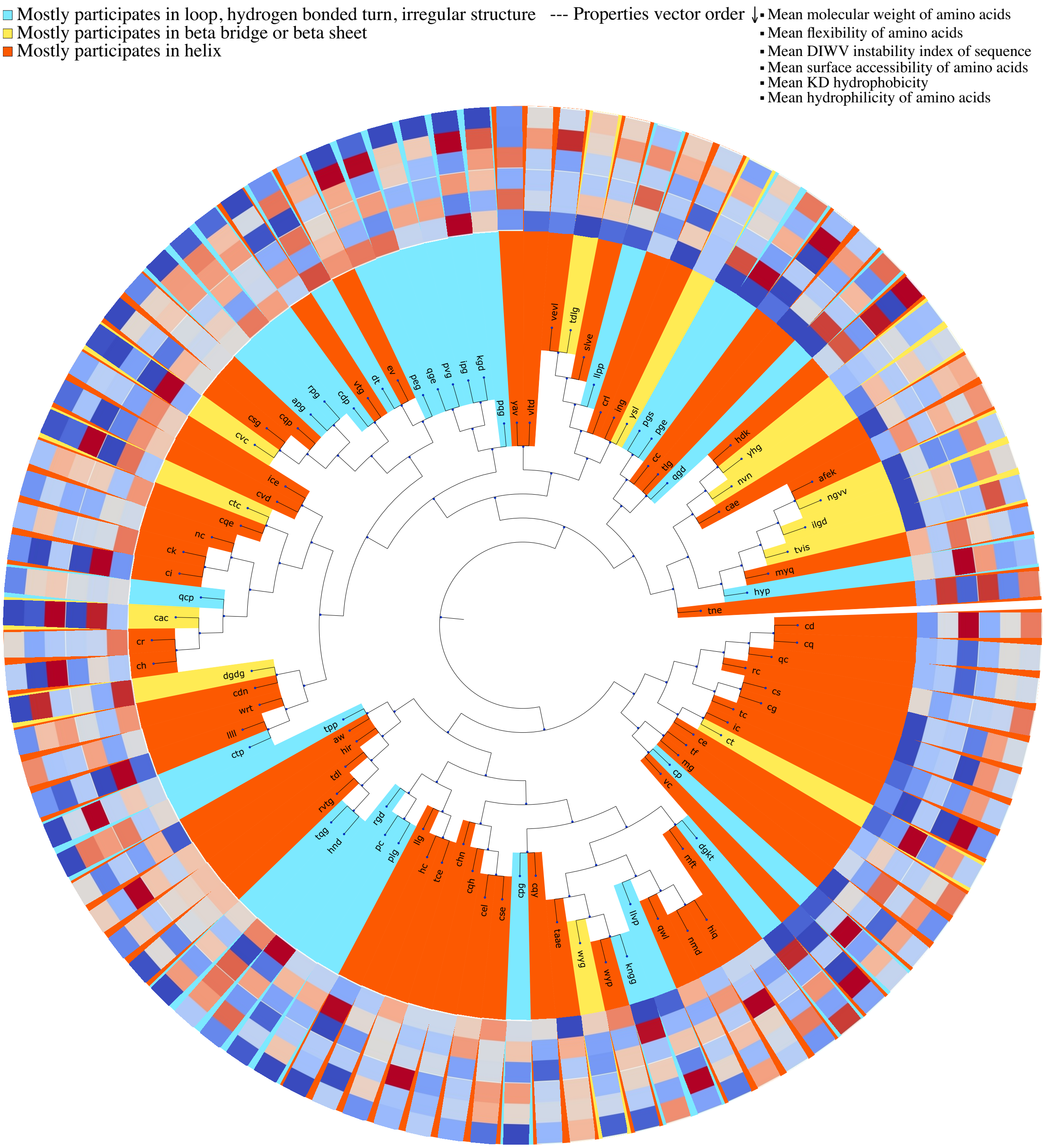


Figure 2.8. Clustering of integrin-binding-specific motifs based on their occurrence in the annotation proteins. This tree diagram illustrates the hierarchical relationships among sequence motifs. On top of the tree structure, the colors visualize different metadata exist for each sequence motif. Each motif is colored based on the most frequent secondary structure it assumes in the Protein Data Bank (PDB) structure out of all PDB sequences. For each motif the biophysical properties are provided in a heatmap visualization, which shows from outer ring to inner ring: the mean molecular weight, mean flexibility, mean instability, mean surface accessibility, mean kd hydrophobicity, and mean hydrophilicity with standardized scores (zero mean and unit variance), where the dark blue is the lowest and the dark red is the highest possible value. Motifs are clustered based on their co-occurrences in the integrin-binding proteins. Zooming in the electronic version is possible for a better see of

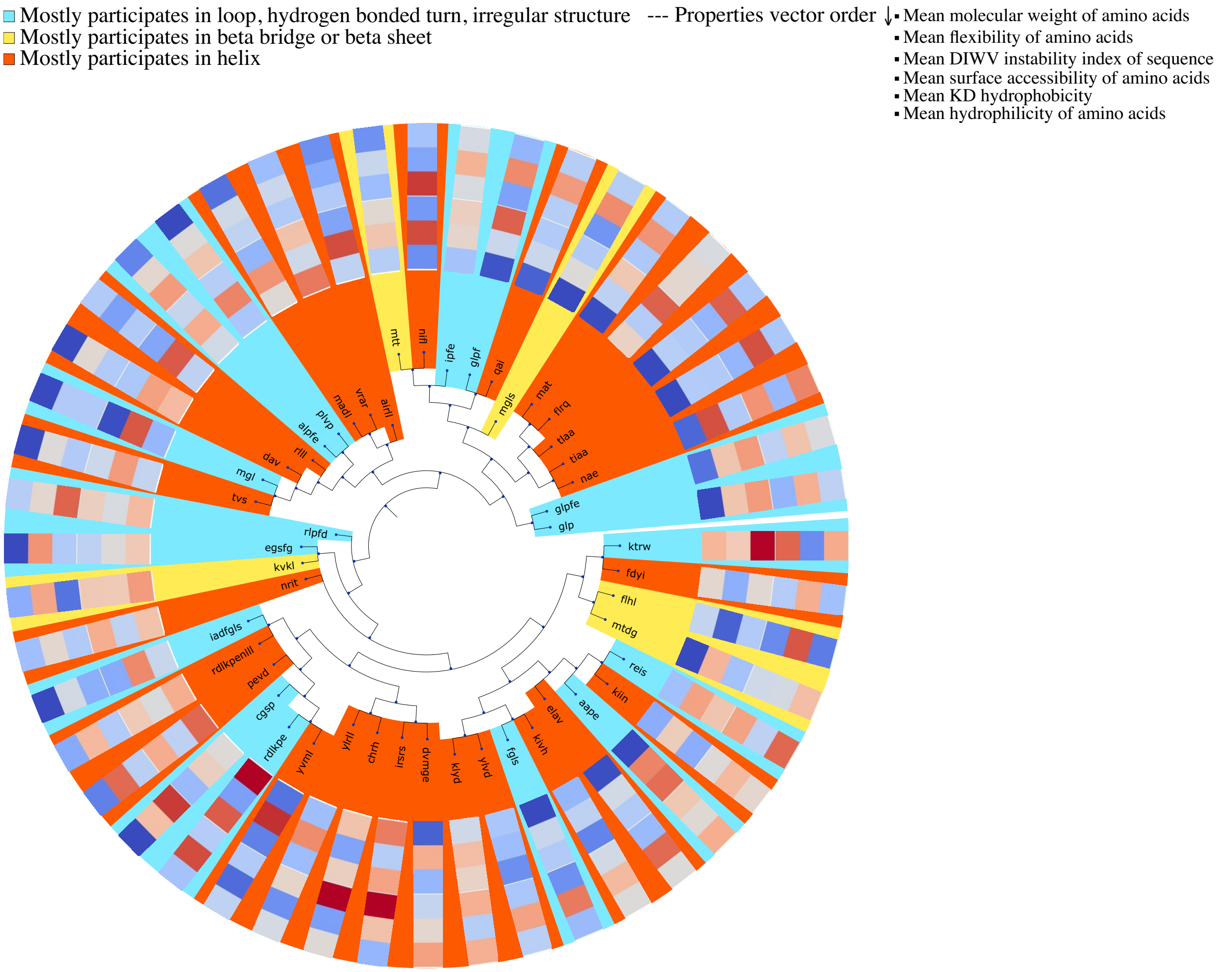


Figure 2.9. Clustering of biofilm formation-specific motifs based on their occurrence in the annotation proteins. Each motif is colored based on the most frequent secondary structure it assumes in the Protein Data Bank (PDB) structure out of all PDB sequences. For each motif the biophysical properties are provided in a heatmap visualization, which shows from outer ring to inner ring: the mean molecular weight, mean flexibility, mean instability, mean surface accessibility, mean kd hydrophobicity, and mean hydrophilicity with standardized scores (zero mean and unit variance), where the dark blue is the lowest and the dark red is the highest possible value. Motifs are clustered based on their co-occurrences in the biofilm formation proteins. Zooming in the electronic version is possible for a better see of the details.

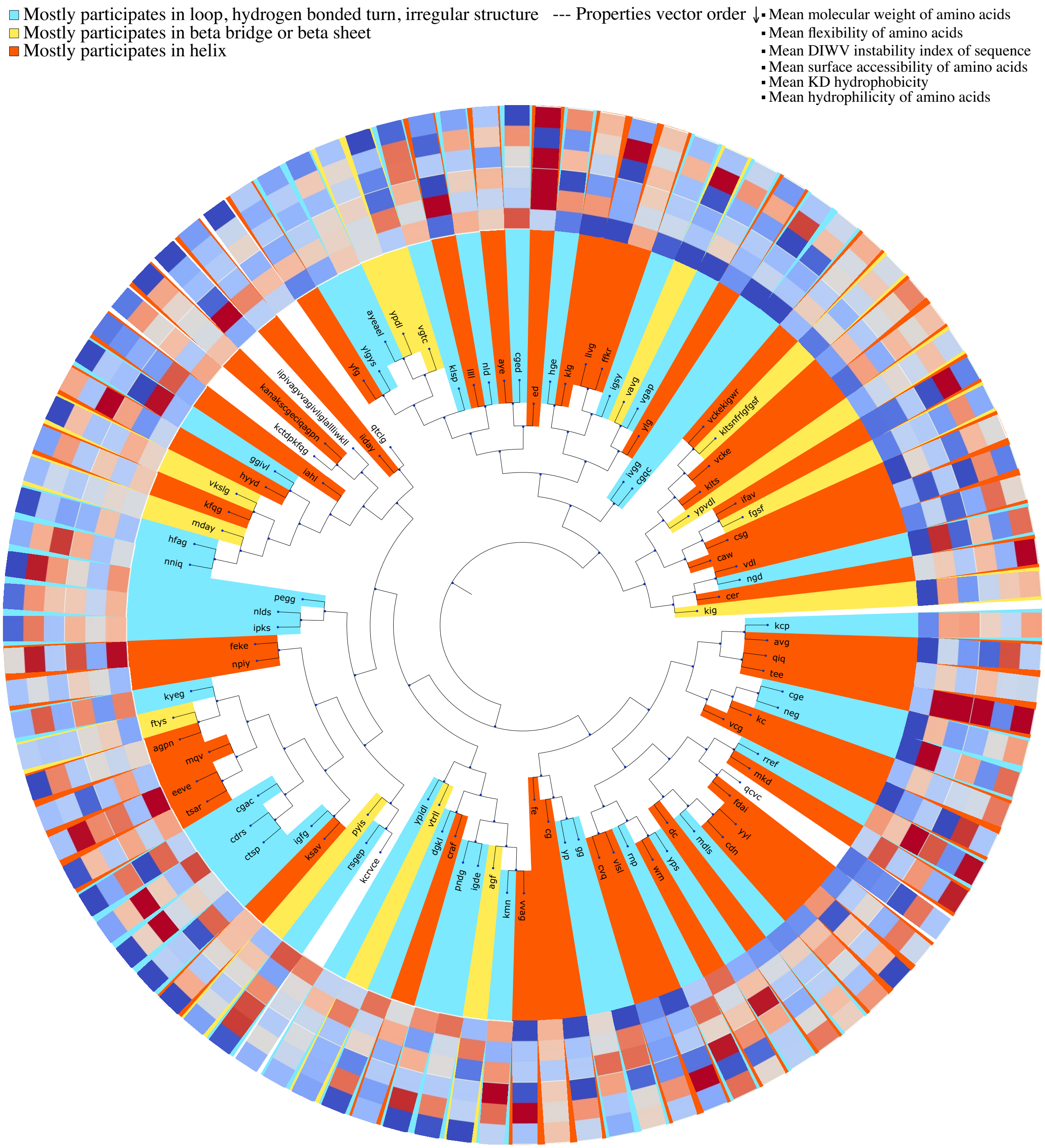


Figure 2.10. Clustering of integrin-related motifs based on their occurrence in the annotation proteins. Each motif is colored based on the most frequent secondary structure it assumes in the Protein Data Bank (PDB) structure out of all PDB sequences. For each motif the biophysical properties are provided in a heatmap visualization, which shows from outer ring to inner ring: the mean molecular weight, mean flexibility, mean instability, mean surface accessibility, mean kd hydrophobicity, and mean hydrophilicity with standardized scores (zero mean and unit variance), where the dark blue is the lowest and the dark red is the highest possible value. Motifs are clustered based on their co-occurrences in the integrin proteins. Zooming in the electronic version is possible for a better see of the details.

Table 2.8. Comparing k-mers, ProtVec, and ProtVecX and their combinations in protein classification tasks. Deep MLP neural network has been used as the classifier.

Dataset	Representation	5 fold cross-validation		
		macro-Precision	macro-Recall	macro-F1
Venom toxin prediction	3-mer	0.89	0.89	0.89
	ProtVec	0.88	0.88	0.88
	ProtVecX	0.88	0.88	0.88
	3-mer + ProtVec	0.90	0.89	0.89
	3-mer + ProtVecX	0.90	0.90	**0.90**
Subcellular location prediction	3-mer	0.65	0.59	0.60
	ProtVec	0.60	0.57	0.58
	ProtVecX	0.57	0.57	0.57
	3-mer + ProtVec	0.68	0.60	**0.62**
	3-mer + ProtVecX	0.66	0.60	0.61
Enzyme prediction	3-mer	0.70	0.73	0.71
	ProtVec	0.68	0.70	0.69
	ProtVecX	0.69	0.71	0.70
	3-mer + ProtVec	0.70	0.73	0.71
	3-mer + ProtVecX	0.71	0.73	**0.72**

vocabulary sizes. This allows for considering multiple ways of segmenting a sequence into sub-sequences. The main purpose of this work was to introduce a variable-length segmentation of sequences, similar to word tokenization in natural languages. In particular, we introduced (i) DiMotif as an alignment-free discriminative protein sequence motif miner, as well as (ii) ProtVecX, a variable-length extension of protein sequence embedding.

We compared DiMotif against two recent existing tools for motif discovery: HH-Motif as an instance of non-discrminative methods, and DLocalMotif as an instance of discriminative methods. We compared the performances in the detection of 20 distinct sub-types of experimentally verified motifs. HH-Motif comparing HMMs of orthologs for retrieving SLiMs, achieved the best average F1 and the DiMotif with domain-specific segmentation achieved the second best F1. DiMotif achieved the highest recall, making it an ideal tool for finding a list of candidates for further experimental verification. Furthermore, we evaluated DiMotif by extracting motifs related to (i) integrins, (ii) integrin-binding proteins, and (iii) biofilm formation. We showed that the extracted motifs could reliably detect reserved sequences of the same phenotypes, as indicated by their high F1 scores. We also showed that DiMotif could reasonably detect experimentally identified motifs related to nuclear localization signals. By using KL divergence between the distribution of motifs in the positive sequences, DiMotif is capable of outputting multi-part motifs. A detailed biophysical interpretation of the motifs is beyond the scope of this work. However, the tree visualization of DiMotif as a tool can help biologists to come up with hypotheses about the motifs for further experiments.

In addition, although homologous sequences in Swiss-Prot have indirectly contributed in DiMotif segmentation scheme, unlike conventional motif discovery algorithms, DiMotif does not directly use multiple sequence alignment information. Thus, it can be widely used in cases motifs need to be found from a set of non-homologous sequences.

We proposed ProtVecX embedding trained on sub-sequences in the Swiss-Prot database. We demonstrated that combining the raw k-mer distributions with the embedding representations can improve the sequence classification performance compared with using either k-mers only or embeddings only. In addition, combining ProtVecX with k-mer occurrences outperformed ProtVec embedding combined with k-mer occurrences for toxin and enzyme prediction tasks. Our results suggest that the recent works in the literature including our previously proposed ProtVec missed serving k-mer representation as a baseline, which is a tough-to-beat baseline. We show that embedding can be used as complementary information to the raw k-mer distribution and their added value is expressed when they are combined with k-mer features.

In this section, we briefly touched motif discovery and protein classification tasks as use cases of peptide pair encoding representation. However, the application of this work is not limited to motif discovery or embedding training, and we expect this representation to be widely used in bioinformatics tasks as general purpose variable-length representation of protein sequences.

2.4 Deep Learning for protein secondary structure prediction

In this section, we investigate deep learning-based prediction of protein secondary structure from the protein primary sequence. We study the function of different features in this task, including one-hot vectors, biophysical features, protein sequence embedding (ProtVec), deep contextualized embedding (known as ELMo), and the Position Specific Scoring Matrix (PSSM). In addition to the role of features, we evaluate various deep learning architectures including the following models/mechanisms and certain combinations: Bidirectional Long Short-Term Memory (BiLSTM), convolutional neural network (CNN), highway connections, attention mechanism, recurrent neural random fields, and gated multi-scale CNN. Our results suggest that PSSM concatenated to one-hot vectors are the most important features for the task of secondary structure prediction. Utilizing the CNN-BiLSTM network, we achieved an accuracy of 69.9% and 70.4% using ensemble top-k models, for 8-class of protein secondary structure on the CB513 dataset, the most challenging dataset for protein secondary structure prediction. Through error analysis on the best performing model, we showed that the misclassification is significantly more common at positions that undergo secondary structure transitions, which is most likely due to the inaccurate assignments of the secondary structure at the boundary regions. Notably, when ignoring amino acids at secondary structure transitions in the evaluation, the accuracy increases to 90.3%. Furthermore, the best performing model mostly mistook similar structures for one another, indicating that the deep learning model inferred high-level information on the secondary structure. The developed software called DeepPrime2Sec and the used datasets and representations are available for further investigations and use by the community at http://github.com/ehsanasgari/deepPrime2Sec.

What is the Protein Secondary Structure?

As discussed in 2.1, protein are important elements for the structure and function of cells. Although a protein sequence should theoretically hold enough information determining its function, finding a mapping from the protein primary sequence to the structure is still one of the open challenges in molecular biology (Hunter 1993; Cooper et al. 2000). The large gap between the number of known protein sequences and the number of known protein 3D structures motivates computational methods and in particular machine learning methods predicting structural information from the protein primary sequences. Protein structure can be described at three main levels: **(i) primary structure** referring to a linear sequence of amino acids, **(ii) secondary structure** referring to the structure of the local segments of the protein sequence categorized into 8 of secondary structures, and **(ii) tertiary structure** referring to the 3D structure of protein macromolecules. In this work, we focused on predicting protein secondary structure from the primary sequences.

Protein secondary structure prediction can be viewed as a sequence labeling machine learning task type, i.e. assigning a categorical label $y_t \in Y$ to each element of a sequence of input elements, $x_t \in X$, where t indicates the position in the sequence. There exist eight possible secondary structure categories (Q8 labeling) for each amino acid at position t in the sequence: the 3-10 helix (G), α helix (H), π helix (I), turn (T), β sheet (E), β bridge (B), bend (S), and loop (L). A simpler labeling scheme is Q3, where the categories are divided into three main classes: helix, strand, and loop/coil. Finding the protein secondary structure is an important step toward the understanding of the protein folding and subsequently its structure and function (Y. Zhou and Karplus 1999; Ozkan et al. 2007). Thus, it can be vital for a variety of protein informatics problem settings, e.g., protein sequence alignment (H. Zhou and Y. Zhou 2005; X. Deng and Cheng 2011), identification of disease-causing mutations (Folkman et al. 2015), and protein function annotation (Taherzadeh et al. 2016).

Secondary structure prediction is one of the primary tasks in protein informatics (Y. Yang et al. 2016). Traditional protein secondary structure prediction include rule-based (Chou and Fasman 1974) methods as well as machine learning approaches using amino acid context-based (Finkelstein and Ptitsyn 1971) and evolution/alignment-based representations (Zvelebil et al. 1987; Hua and Sun 2001). Recently, with the popularity of neural network approaches, similar to many machine learning prediction tasks, different neural network architectures have been proposed for the protein secondary structure prediction from the primary sequences. To the best of our knowledge, all of these architectures have been mainly used on top of a set of fixed features (PSSM and one-hot vector) and mostly on a fixed predictive model in each work. The most challenging dataset for this task has been the 8-way secondary structure prediction using the CullPDB dataset as training and CB513 as the test set (Sonderby and Winther 2014). Different deep learning architectures proposed for this task includes: (i) deep Convolutional Generative Stochastic Network, obtaining a test accuracy of 66.4% (Jian Zhou and Troyanskaya 2014), (ii) Long Short-Term Memory (LSTM) recurrent neural network, obtaining test accuracy of 67.4% (Sonderby and Winther 2014), (iii) Convolutional neural fields achieving a test accuracy of 68.3% (S. Wang et al. 2016), (iv) bi-directional LSTM with/without conditional random field (CRF) (Johansen et al. 2017; Jurtz et al. 2017), obtaining a test accuracy of 69.4% and 68.5% respectively, (v) gated Multi-scale convolutional neural network (multi-scale CNN) (Jiyun Zhou et al. 2018) with a test accuracy of 69.3% and 70.3%.

The main contributions of the present chapter are 4-fold, namely (i) we provide a systematic comparison of representations that can be used for the secondary structure prediction task. (ii) We provide a comparison of several important deep learning architectures for this task in order to find the best performing model. (iii) we provide the community with a framework for advancing deep learning techniques in the area of secondary structure prediction, called DeepPrime2Sec (iv) We perform a detailed analysis of the location and classes of errors.

We first implement one of the state-of-the-art neural architecture for sequence labeling tasks in different domains (e.g., proteomics (Jurtz et al. 2017), natural language processing (Lample et al. 2016)). Subsequently, we investigate the role of different features in the performance of protein secondary structure prediction. In particular, we utilize five sets of features and their combinations: (i) one-hot vector representation, (ii) biophysical scores of amino acids, (iii) amino-acid protein vectors, (iv) recently introduced contextualized embeddings, and (v) Position-Specific Scoring Matrix (PSSM). Secondly, for the best feature set in the preceding step, we investigate the performance of different deep learning architectures for the task of secondary structure, including convolutional-recurrent neural network (with and without conditional random field layer), highway connections, attention mechanism, and multi-scale CNN models. Our results confirm that PSSM is the most informative feature for the protein secondary structure, and other features only result in slight improvements when they are combined with PSSM. Error analysis indicates that errors are mostly occurring at positions were transitions between successive secondary structure elements occur. We provide the Prime2Sec code for further investigations.

Materials and Methods

Datasets

Secondary structure prediction dataset: Several benchmark datasets exist in the literature of protein secondary structure prediction. The most challenging dataset based on the maximum achieved accuracy using machine learning approaches is the CullPDB dataset, which consists of 5,534 protein sequences (CullPDB-train) for training and 513 non-redundant sequences (CB513) for test purpose (Jian Zhou and Troyanskaya 2014), in which sequences with more than 25% sequence identity to training data entries were removed from the validation set.

The label for each position is selected from the Q8 scheme (explained in § 2.4). We used as performance metric accuracy, which is the most common metric for the evaluation of protein secondary structure predictors over the filtered CB513. Accuracy can be defined as the ratio of correctly predicted secondary structures in amino acid level.

UniRef50 dataset for ELMo training: We use UniRef50 collection as the training data to learn the contextualized embeddings. The primary purpose of using this dataset instead of the whole Swiss-Prot or UniProt has been to avoid having redundant sequences in the test set. We use 90% of sequences for training and 10% for test purpose.

Swiss-Prot for ProtVec training: As we proposed in (Asgari and M. R. Mofrad 2015) for the training of ProtVec embedding, we use the whole Swiss-Prot dataset containing 600Kprotein sequences.

Approach

First, to investigate the effect of different representations on secondary structure prediction performance, we fix the predictive model and investigate the accuracy under representation changes. For this purpose, we re-implement the state-of-the-art architecture for sequence labeling tasks, i.e., convolutional bidirectional LSTM model (Johansen et al. 2017). The general architecture used for secondary structure in this paper is illustrated in 2.11 (a). Secondly, we examine the impact of several deep learning architectures on top of the best feature set obtained in the first step. Thirdly, we create an ensemble predictor on the best performing models. Finally, we present an error analysis of the misclassification locations and confusing secondary structure categories. The experiment steps are detailed as follows.

Investigation on the contribution of features in protein secondary structure prediction

We experiment on five sets of protein features to understand what are essential features for the task of protein secondary structure prediction. Although in 1999, PSSM was reported as an important feature to the secondary structure prediction (Jones 1999), this was still unclear whether recently introduced distributed representations can outperform PSSM in such a task. For a systematic comparison, the features detailed as follows are used:

- **One-hot vector representation (length: 21)**: vector representation indicating which amino acid exists at each specific position, where each index in the vector indicates the presence or absence of that amino acid.

- **ProtVec embedding (length: 50)**: representation trained using Skip-gram neural network on protein amino acid sequences(Asgari and M. R. Mofrad 2015), detailed in §2.4. The only difference would be character-level training instead of n-gram based training.

- **Contextualized embedding (length: 300)** : we use the contextualized embedding of the amino acids trained in the course of language modeling(Peters et al. 2018), known as ELMo, as a new feature for the secondary structure task. Contextualized embedding is the concatenation of the hidden states of a deep bidirectional language model. The main difference between ProtVec embedding and ELMO embedding is that the ProtVec embedding for a given amino acid or amino acid k-mer is fixed and the representation would be the same in different sequences. However, the contextualized embedding, as it is clear from its name, is an embedding of word changing based on its context. We train ELMo embedding of amino acids using UniRef50 dataset in the dimension size of 300.

- **Position Specific Scoring Matrix (PSSM) features (length: 21)**: PSSM is amino acid substitution scores calculated on protein multiple sequence alignment of homolog sequences for each given position in the protein sequence.

- **Biophysical features (length: 16)** For each amino acid we create a normalized vector of their biophysical properties, e.g., flexibility (Vihinen et al. 1994), instability (Guruprasad et al. 1990), surface accessibility (Emini et al. 1985), kd-hydrophobicity (Kyte and Doolittle 1982), hydrophilicity (Hopp and Woods 1981), and etc.

Feature combinations: We start from using single features and then greedily use the best combinations. Since the PSSM features have a significant contribution in improving the accuracy, we select then for every combination of length two feature sets and more. We perform an extensive parameter tuning of the network hyper-parameters (LSTM size, CNN filter sizes, etc.) for each feature combination.

Data augmentation for the best set of features: as later will the reader will see in the results, the combinations of one-hot and PSSM lead to the best performance on the test set. Since PSSM has information about the possible substitution of amino acids in the homolog sequences and homologous sequences are likely to have similar functions and structures, we come up with the idea of generating more data points using PSSM feature (based on most possible mutations) and test it as a data-augmentation policy. This way, we produce ten possible samples from each training instance by changing the amino acid (or more accurately keeping PSSM the same and change the one-hot by sampling from PSSM vector).

Deep learning models for protein secondary structure prediction

In addition to investigation on the relevant features, we use different deep learning architectures on top of the selected feature in §2.4 and examine the role of architecture for the protein secondary structure prediction. In a general notation, each sequence labeling task can be viewed as assigning a sequence of labels $\mathbf{Y} = (y_1, y_2, \dots, y_T)$ to the chain of elements in a given input sequence $\mathbf{X} = (x_1, x_2, \dots, x_T)$. Using neural architectures, we look for $\mathbf{Y}^*$ maximizing $P_\theta(\mathbf{Y}|\mathbf{X})$. P_θ is parameterized using a softmax function:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{\exp\left(\mathrm{S}(\mathbf{X}, \mathbf{Y})\right)}{\sum_{\mathbf{Y}'} \exp\left(\mathrm{S}(\mathbf{X}, \mathbf{Y}')\right)},$$

where S is a score function computed through a neural network relating the input $\mathbf{X}$ to the output $\mathbf{Y}$. The difference between different models is in the neural architecture parametrizing the S. We study the use of various deep learning architectures to produce S described as follows.

(a) CNN-BiLSTM Model: As the CNN-BiLSTM model is illustrated in Figure 2.11 (a), firstly convolutional filters of different window sizes are applied on the input features, creating feature maps of different neighborhoods. Then the feature maps are concatenated. The resulting vector encodes the representation of different context sizes around each amino acid in the protein sequence. Batch normalization is used to increase the stability of training (Ioffe and Szegedy 2015). Subsequently, a fully-connected neural network projects the result into a dense vector. In order to avoid over-fitting, dropout is used (N. Srivastava et al. 2014a). Up to this point we encoded a sequence of amino acid features vectors $\mathbf{X} = (x_1, x_2, \dots, x_T)$ into a dense vector $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_T)$ using a convolutional and feedforward neural network, which is enriched on local information at each position t. However, v_t does not encode global information about the sequence. A long Short-term Memory Network (LSTM) (Hochreiter and Schmidhuber 1997), which is designed to capture long-range dependencies, encodes the sequence information. The LSTM creates an encoding $\overrightarrow{\mathbf{h}_t}$ of the left context of the protein at position t. Since both left and right contexts can be crucial for the global structure of proteins, we use a bi-directional LSTM (or shortly BiLSTM network). The Bi-LSTM encodes each position into a representation of left and right contexts $\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t]$. Utilizing feedforward layers (with dropout) on top of $\mathbf{h}_t$, we create a vector Φ with a length of the number of possible target labels ($|Y| = 8$). Φ_t can be regarded as $S(x_t, y)$. Applying a softmax function on the score vector S, the label with the maximum value in S is chosen.

(b) CNN-BiLSTM with Highway Connections: the importance of PSSM features in the secondary structure prediction motivates a more direct application of this representation in the last layer as a highway connection (K. He et al. 2016). We concatenated the output of BiLSTM $\mathbf{h}_t$ with the highway connection to batch-normalized PSSM feature.

(c) CNN-BiLSTM with Conditional Random Field (CRF) layer: in some natural language processing sequence labeling tasks where there is a complex dependency between neighbor labels, e.g., named entity recognition tasks, using a conditional random field layer on top of the recurrent neural network enhances the prediction power by adding constraints to the final predicted labels (Lample et al. 2016). A similar idea has been applied in (Johansen et al. 2017) to the protein secondary structure prediction. In this model, instead of parametrizing the S only with the output of the Bi-LSTM model, we use a CRF layer to consider the transition probability between neighboring labels as well:

$$\mathrm{S}(\mathbf{X}, \mathbf{Y}) = \sum_t \log \Phi(y_t, x_t) + \log \psi(y_{t-1} \to y_t),$$

where Φ comes from the output of the BiLSTM and the subsequent feedforward layer and the $\psi(y_{t-1} \to y_t)$ is the potential function in the CRF model.

d) CNN-BiLSTM with Attention layer: another approach for defining S is to write it as the weighted average of all LSTM hidden states. This way, we would allow the model to benefit from the weighted long-term dependencies (global context of the current amino acid). The modification is shown in Figure 2.11 (d):

$$h_{tj} = tanh(h_t^T W_t + h_j^T W_c + b_t)$$

$$e_{tj} = \sigma(h_{tj}^T W + b)$$

$$a_{tj} = \frac{\exp(e_{tj})}{\sum_{j \in \mathbf{T}} \exp(e_{tj})}$$

$$c_t = \sum_{j \in \mathbf{T}} a_{tj} h_{tj}$$

$$S_t = ReLU(c_t^T W_s + b_s)$$

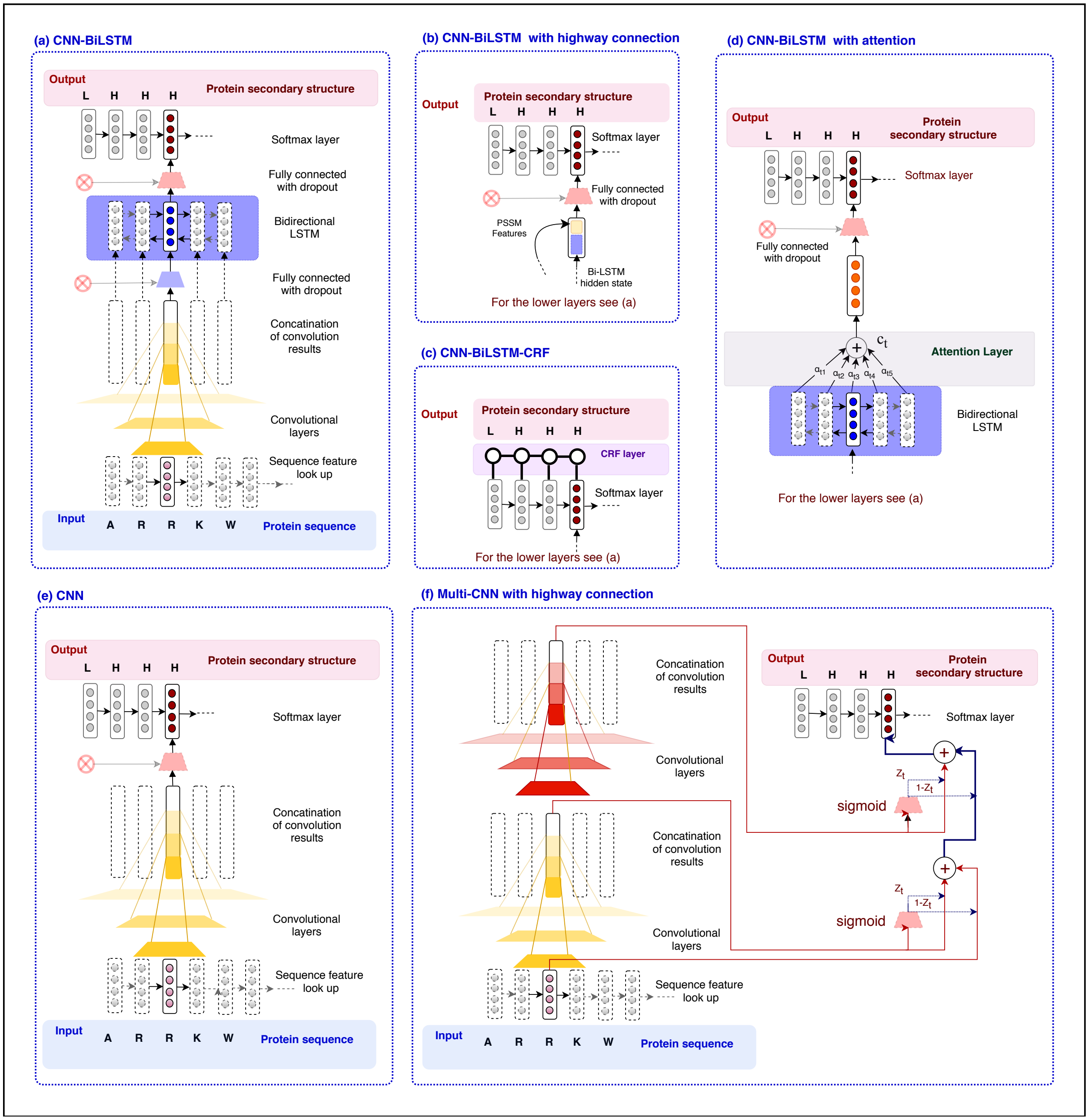

(a) CNN-BiLSTM
Output
Protein secondary structure
L H H H
Softmax layer
Fully connected with dropout
Bidirectional LSTM
Fully connected with dropout
Concatination of convolution results
Convolutional layers
Sequence feature look up
Input
A R R K W
Protein sequence
(b) CNN-BiLSTM with highway connection
Output
Protein secondary structure
L H H H
Softmax layer
Fully connected with dropout
PSSM Features
Bi-LSTM hidden state
For the lower layers see (a)
(c) CNN-BiLSTM-CRF
Output
Protein secondary structure
L H H H
CRF layer
Softmax layer
For the lower layers see (a)
(d) CNN-BiLSTM with attention
Output
Protein secondary structure
L H H H
Softmax layer
Fully connected with dropout
c_t
Attention Layer
α_{t1} α_{t2} α_{t3} α_{t4} α_{t5}
Bidirectional LSTM
For the lower layers see (a)
(e) CNN
Output
Protein secondary structure
L H H H
Softmax layer
Concatination of convolution results
Convolutional layers
Sequence feature look up
Input
A R R K W
Protein sequence
(f) Multi-CNN with highway connection
Concatination of convolution results
Convolutional layers
Output
Protein secondary structure
L H H H
Softmax layer
Z_t
$1-Z_t$
sigmoid
Concatination of convolution results
Convolutional layers
Z_t
$1-Z_t$
sigmoid
Sequence feature look up
Input
A R R K W
Protein sequence

Figure 2.11. Different deep learning architectures implemented and evaluated in DeepPrime2Sec are provided as follows: **(a)** CNN-BiLSTM, The central neural architecture we used for protein secondary structure; **(b)** The modification of the model (a) with adding highway connection to the CNN-BiLSTM for a more direct use of PSSM features; **(c)** The modification of the model (a) by adding a CRF layer to consider label consistency in the prediction; **(d)** The modification of the model (a) by adding an attention layer to the output of LSTM for considering a more global context; **(e)** Solely using convolutional layer **(f)** Multiscale convolutional neural network benefiting from a gating mechanism (as proposed in (Jiyun Zhou et al. 2018)).

where $h_{t,j}$ is the contribution of each previous and future LSTM steps in the current state, and $a_{t,j}$ is the normalized step's contributions (attention weight), which is used to weight the LSTM $h'_x s$. The final encoding for the time t would be c_t instead of h_t. Finally a non-linear transformation of c_t creates a vector in size of the targets (8 classes) representing the score function S.

(e) CNN: An alternative architecture for this task would be the pure use of the convolutional neural network on the sequence axis (the order in the sequence) to capture local information for prediction on the secondary structure.

(f) Multiscale-CNN with a highway connection: we implemented one of the state-of-the-art architectures proposed in (Jiyun Zhou et al. 2018) as part of *DeepPrime2Sec*, which uses a gated version of stacked CNNs. The ultimate output of each convolutional layer would be a gated version of its input and the convolutional result as depicted in Figure 2.11 (f):

$$o_t = z_t * c_t + (1 - z_t) * o_{t-1}$$
$$z_t = \sigma(o_{t-1}^T W_z + b)$$
$$o_0 = input,$$

where c_t is the result of the t^{th} convolutional layer, o_t is the out of gating between the previous and the current convolutions. In the first layer, O_0 is the same as the input layer. **Ensemble of the best models:** similar to previous work, to improve this performance further using "Wisdom of Crowds principle", we produced an ensemble classifier on the top-k classifiers (k=5,10,20,50,100) and predict on the test set by voting.

Results

Results on the role of features in secondary structure prediction

Single features: The performance of protein secondary structure prediction using different combinations of features is provided in Table 2.9. When we used single feature sets, PSSM features performed substantially better than other features. Even recent deep learning-based representations were far behind. PSSM is amino acid substitution scores calculated on protein multiple sequence alignment of homolog sequences for each given position in the protein sequence. This result confirms that evolution has very important information for defining the protein structure. One-hot vector encoding performed similar to other amino-acid embedding approaches (ELMo, ProtVec, and biophysical features). The reason behind can be that the one-hot vectors and protein embeddings are acting complementary to each-other; one-hot vector representation increases the precision about the specific residue at position t, while embedding blurs this information by providing information about the context of this amino acid. Among the embedding methods, ELMo embedding worked marginally better than other approaches, as it has information about long-term past and future of the sequence, i.e., having information on a more the global context) in comparison to ProtVec, which only includes information about a local context within a certain context size.

Combination of features: The second set of rows in Table 2.9 shows the combinations of features with PSSM (the best performing feature) starting from combinations of two types up to five feature types. We also optimized the hyperparameters (network sizes and parameters). The combination of one-hot and PSSM turned out to be the most effective representation, while using combination of features did not substantially improve the accuracy further. A reason could be that the PSSM implicitly already includes information about the biophysical and contextual features, at least as much as needed for the protein secondary structure prediction, more than other embedding approaches. Further tuning of the convolution window sizes resulted in the accuracy of 69.9% (Table 2.9).

Data augmentation: Although augmenting the dataset by keeping the PSSM and generating more one-hots made the training dataset 10x larger; the performance could not be improved further 2.9. Next, we generate ten instances from each test instance and used the best model to predict 10 secondary structures for each position, based on the augmented test set and take the majority vote. This idea also did not boost performance. We conclude that the most important feature to this task is the PSSM, and we need to find a way to perform augmentation on this feature for further performance improvements.

Results on comparison of deep learning architectures in protein secondary structure prediction

The results on secondary structure prediction for different deep learning architectures using the best feature set (i.e., the combination of PSSM and one-hot representation, see §2.4) is provided in Table 2.11. Using only BiLSTM and only CNN, we could achieve accuracies of 67.1% and 68.1%, respectively. Combination of CNN and BiLSTM led to the best observed accuracy of 69.9%. Adding the CRF layer, attention layer, and the highway connection did not further improve the performance. From this, we may conclude that the LSTM hidden states stored sufficient information, ensuring to provide a logical sequence of labels. The previously reported performances on the *CB513* are also provided in Table 2.11. The CNN-BiLSTM architecture outperformed all, except for the *CNNH_PSS* model. Based on the provided implementation and descriptions in the (Jiyun Zhou et al. 2018), we attempted to reproduce the architecture and the results in the *DeepPrime2Sec*. However, using this architecture, we could not obtain better accuracy than 66.9%.

Ensemble predictor: To further improve this performance, we produced an ensemble classifier on top-k classifiers (k=5,10,20,50,100) resulting in the accuracy of 70.4 (Table 2.9) outperforming the 70.3 the state-of-the-art performance (Jiyun Zhou et al. 2018).

Table 2.9. Protein secondary structure prediction results using (i) different feature types, (ii) their combinations, (iii) extensively tuned hyperparameters for the best feature sets, (iv) the data augmentation are presented.

Representation	Convolution filters	LSTM input/hidden/output	CB513 Q8 accuracy
(i) Single Features			
Biophysical features (16D)	$[3,5,7],16x$	400 - 800 - 400	0.573
ELMo embedding (300D)	$[3,5,7],128x$	400 - 800 - 400	0.577
ProtVec embedding (50D)	$[3,5,7],128x$	400 - 800 - 400	0.573
PSSM (21D)	$[3,5,7],128x$	400 - 800 - 400	0.694
One-hot (21D)	$[3,5,7],16x$	200 - 400 - 200	0.575
(ii) Combinations of top features			
Biophysical features & PSSM	$[3,5,7],256x$	1000 - 1000 - 1000	0.697
ELMo & PSSM	$[3,5,7],256x$	400 - 800 - 400	0.692
ProtVec embedding & PSSM	$[3,5,7],256x$	1000 - 1000 - 1000	0.696
One-hot & PSSM	$[3,5,7],256x$	1000 - 1000 - 1000	**0.698**
Biophysical features & ELMo & PSSM	$[3,5,7],256x$	400 - 800 - 400	0.692
Biophysical features & ProtVec embedding & PSSM	$[3,5,7],256x$	500 - 1000 - 500	0.694
Biophysical features & one-hot & PSSM	$[3,5,7],256x$	2000 - 1000 - 2000	**0.699**
ELMo & ProtVec embedding & PSSM	$[3,5,7],256x$	400 - 800 - 400	0.692
ELMo & one-hot & PSSM	$[3,5,7],256x$	1000 - 1000 - 1000	0.692
ProtVec embedding & one-hot & PSSM	$[3,5,7],128x$	400 - 800 - 400	0.694
Biophysical features & ELMo & ProtVec embedding & PSSM	$[3,5,7],256x$	1000 - 1000 - 1000	0.69
Biophysical features & ELMo & one-hot & PSSM	$[3,5,7],256x$	500 - 1000 - 500	0.693
Biophysical features & ProtVec embedding & one-hot & PSSM	$[3,5,7],256x$	500 - 1000 - 500	0.694
ELMo & ProtVec embedding & one-hot & PSSM	$[3,5,7],256x$	500 - 1000 - 500	0.692
Biophysical features & ELMo & ProtVec embedding & one-hot & PSSM	$[3,5,7],256x$	1000 - 1000 - 1000	0.692
(iii) Parameter tuning for the selected features			
One-hot & PSSM	$[3,5,7,11,21],256x$	1000 - 1000 - 1000	**0.699**
(iv) Data augmentation for the selected parameter setting			
Augmented one-hot & PSSM in training	$[3,5,7,11,21],256x$	1000 - 1000 - 1000	0.692
Augmented one-hot & PSSM in testing	$[3,5,7,11,21],256x$	1000 - 1000 - 1000	0.68

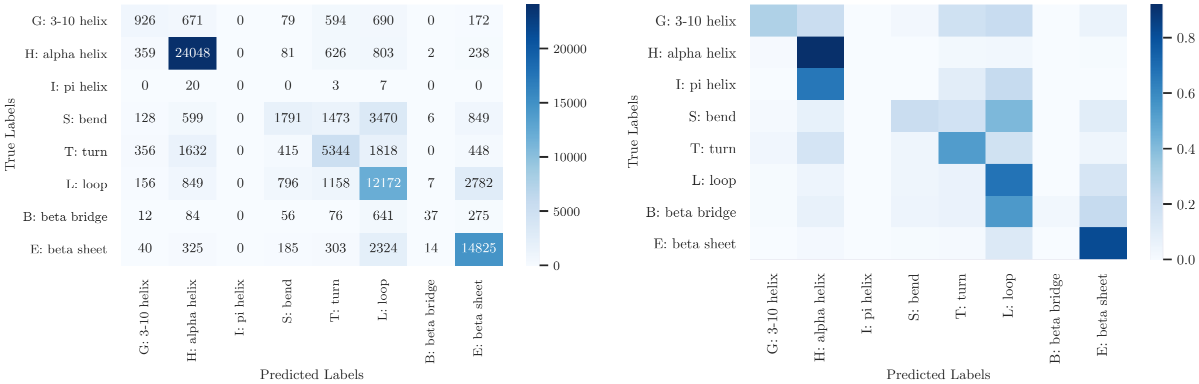
Location analysis of misclassified amino acids: Predicting the secondary structure of amino acids at the transition points (transition between two distinct secondary structure) can be tricky. We had the hypothesis that transition positions may have a larger chance of misclassification, as even the ground-truth quality can be lower in these states (Y. Yang et al. 2016). To study the effect of amino acid position in the misclassifications, we performed statistical tests to determine whether the misclassification event is dependent or independent of locating to the boundaries of the secondary structures in the $CB513$ target labels. We performed both χ^2 and log-likelihood ratio (i.e., the G-test) tests and found that the misclassification highly depends on being at the transition location (the p-values on both χ^2- and G-tests were ≈ 0). The underlying contingency table is shown in Table 2.10. Surprisingly, if we omit amino-acids at the borders from the evaluation, the Q8 accuracy would increase to %90.3.

A categorical analysis of the misclassified amino acids: The confusion matrix and $l1$ normalized confusion matrix for the best performing model using the combination of PSSM and one-hot vector in CNN-BiLSTM architecture are presented in Figure 2.12 (a) and (b) respectively. Since the protein secondary structure problem setting is relatively imbalanced,

Table 2.10. Contingency table for location analysis of the misclassified amino acids

		Located at the PSS* transition	
		True	**False**
misclassified	**True**	# 22615	# 3002
	False	# 31156	# 27992

the normalized confusion matrix in Figure 2.12 (b) can more clearly show which classes are relatively confused with each other. The most common secondary structure classes of the alpha helix (H), loop (L), beta sheet (E) were predicted accurately. The classes of bend (B), turn (T) were considerably confused with loop (L), which makes sense, as these classes are similar to each-other and loop (L) is the most frequent one among them. As expected based on structural similarities, bend (S), turn (T), and loop (L) as well as 3-10 helix (G) and alpha helix (H) were also highly confused.



(a) Confusion matrix of the best protein secondary structure prediction model

(b) Normalized confusion matrix of the best protein secondary structure prediction model

Figure 2.12. The confusion matrices of the best performing predictor for protein secondary structure prediction on the test data (on CB513) are provided. Figure (a) shows the standard confusion matrix. However, since the secondary structures are not balanced, for better visualization class confusions, we have $l1$ normalized the rows in figure (b).

Discussions and Conclusion

We studied the machine learning-based protein secondary structure prediction approaches from the protein primary sequence. We focused on finding an optimal representation and deep learning predictive model for this task. The most challenging dataset for this task to-date is Q8 (8 classes) on CullPDB/CB513 dataset, where the dissimilarity of training and

Table 2.11. Protein secondary structure prediction results on different deep learning architectures implemented in DeepPrime2Sec, on top of the combination of PSSM and one-hot representation and the ensemble of their top-k models are shown and compared to the state-of-the-art approaches on the CB513 test set.

DeepPrime2Sec Neural networks	**Q8 accuracy - CB513**
CNN	68.1%
BiLSTM	67.5%
CNN-BiLSTM	**69.9%**
CNN-BiLSTM-CRF	69.0%
CNN-BiLSTM with highway connection	69.6%
CNN-BiLSTM with attention layer	69.2%
Muliscale CNN with highway connection	66.9%
Ensemble of top neural networks/features	
Ensemble of 5 neural networks	70.3%
Ensemble of 10 neural networks	70.2%
Ensemble of 20 neural networks	70.3%
Ensemble of 50 neural networks	70.3%
Ensemble of 100 neural networks	**70.4%**
Previously reported performances in the literature	
Zhou & Troyanskaya (Jian Zhou and Troyanskaya 2014) 2014 (CNN)	66.7%
Sønderby & Winther (Sonderby and Winther 2014) 2014 (LSTM)	67.4%
Wang et al (S. Wang et al. 2016) 2016 (Convolutional neural fields)	68.3%
Johansen et al (Johansen et al. 2017) 2017 (biRNN-CRF)	69.4%
Johansen et al (Johansen et al. 2017) 2017 (biRNN)	68.5%
Zhou et al (Jiyun Zhou et al. 2018) 2018 (Multi-scale CNN)	69.3%
Zhou et al (Jiyun Zhou et al. 2018) 2018 (Multi-scale CNN)	**70.3%**

test set is ensured. We investigated (i) different protein sequence representations including one-hot vectors, biophysical features, protein sequence embedding (ProtVec), deep amino acid contextualized embedding (ELMo), and the Position Specific Scoring Matrix (PSSM), (ii) different deep-learning architectures including convolutional neural networks (CNN), recurrent neural networks (in particular Bi-LSTM), use of highway connection, attention mechanism, and multi-scale CNN (). We show that PSSM and its combination with one-hot vectors achieves the best performance in protein secondary structure prediction. The best performing model was the CNN-BiLSTM architecture, which captures both local and global sequence features essential for proteins secondary structure. We explored data augmentation of one-hot vector based on the PSSM, which was not successful. A future direction could be exploring possible data augmentation schemes of PSSM features.

DeepPrime2Sec provides the community with a deep learning tool specialized for the protein secondary structure prediction covering different architectures. The BiLSTM-CRF architecture performs competitive to the other existing approach in the literature, and the ensemble of the best performing model in Prime2Sec marginally outperforms the existing methods.

In addition, we performed error analysis on the most accurate model based on the location of misclassified amino acids as well as the confusion matrix analysis. Strikingly, misclassified secondary structures were significantly correlated with locating at the structural transitions. Such a correlation is most likely due to the inaccurate assignment of the secondary structure at the boundaries in ground-truth (Y. Yang et al. 2016). By ignoring the boundary amino acids from the evaluation, the Q8 accuracy would increase for an extra %20, i.e., %90.3. Analysis of the confusion matrix furthermore indicates that similar secondary structures are highly confusing (helices: H and G as well as unstructured regions: S, T, and L) showing that the model can learn high-level information about the secondary structures. Even if the exact secondary structure was not predicted, the predicted structure is similar to the target structure.

2.5 Summary of contributions in proteomics

We conclude the proteomics chapter with a summary of contributions of this dissertation in protein informatics:

- Introducing a language model based distributed representation of protein sequences (ProtVec)
- The probabilistic variable-length segmentation of protein sequences for both motif mining and extension of the ProtVec to variable-length embeddings
- Providing state-of-the-art protein secondary structure predictor from the primary sequences through a comprehensive investigation on the role of representations and deep learning architectures in this task.

Distributed representation of protein sequences

The large gap between the number of known protein sequences (raw data) versus the number of known functions/structures associated with these sequences (meta-data) motivates developing methods that can obtain prior knowledge from the existing raw sequences to infer information about structure and function of protein sequences. Continuous vector representations of words known as word vectors have recently become popular in natural language processing (NLP) as an efficient unsupervised approach to represent semantic and syntactic units of text helping in the NLP tasks (e.g., machine translation, parsing, part-of-speech tagging, information retrieval, etc.). Inspired by this idea, we proposed distributed vector representations of biological sequence segments (k-mers), called bio-vector in general and ProtVec for proteins, using the skip-gram neural network. We proposed an intrinsic evaluation of ProtVec by measuring the continuity of the underlying biophysical properties (e.g., average mass, hydrophobicity, charge, and etc.) using the best Lipschitz constant. In addition to intrinsic evaluations, for extrinsic evaluations, we evaluated ProtVec representation in the classification of 324,018 protein sequences belonging to 7,027 protein families, where an average family classification accuracy of $93\% \pm 0.06\%$ was obtained. In addition, incorporation of this representation versus one-hot vector features in Max-margin Markov Network (M^3Net) for intron-exon prediction and domain identification tasks could improve the sequence labeling accuracy from 73.84% to 74.99% and from 82.4% to 89.8%, respectively.

To the best of our knowledge, for the first time, we introduced language model-based embedding of biological sequences. In particular, we used it for unsupervised feature extraction from protein sequences for down-stream machine learning on protein structural and functional annotation. ProtVec has been used and extended for variety of tasks in bioinformatics, where we can only cite a subset of such papers (Wan and J. Zeng 2016; Islam et al. 2017; Hamid and Friedberg 2018; K. K. Yang et al. 2018; Jaeger et al. 2018; Du et al. 2018; A. Dutta et al. 2018; Y. Xu et al. 2018; Öztürk et al. 2018). In addition to contributions of ProtVec in bioinformatics, similar approaches later were introduced for subword embedding in

NLP (Schütze et al. 2016; Kocmi and Ondrej Bojar 2016), where some were directly inspired by ProtVec work (Schütze et al. 2016).

We extended ProtVec embedding to ProtVecX (extended ProtVec) trained on sub-sequences in the Swiss-Prot database (detailed in Section §2.3). We demonstrated that combining the raw k-mer distributions with the embedding representations (either of ProtVec to ProtVecX) can improve the sequence classification performance compared with using either k-mers only or embeddings only. In addition, combining ProtVecX with k-mer occurrences outperformed ProtVec embedding combined with k-mer occurrences for toxin and enzyme prediction tasks. These results suggest that embedding can be used as complementary information to the raw k-mer distribution and their added value is expressed when they are combined with k-mer features. Using the same representation (the combination of embeddings and k-mer features) as an input to a deep neural network, we achieved the first and the third places in two out of three protein classification tasks in the Critical Assessment of protein Function Annotation (CAFA) in 2018 (CAFA 3.14) (N. Zhou et al. 2019).

Probabilistic variable-length segmentation of protein sequences

One of the obvious differences between biological sequences and many natural languages is that biological sequences (DNA, RNA, and proteins) often do not contain clear segmentation boundaries of the sequence segments. This difference makes the fixed length k-mer subsequences the common unsupervised approach in bag-of-word representation of biological sequences, including proteins. However, these fixed-length k-mers can be arbitrary units without any biological implication. Thus, more meaningful units need to be introduced. We proposed a new unsupervised method for segmentation of protein sequences. Instead of fixed-length k-mers, we segmented sequences into the commonly occurring variable-length sub-sequences, inspired by BPE, a data compression algorithm. These sub-sequences were then used as input features to the learning algorithm. As a modification to the original BPE algorithm, we defined a probabilistic segmentation by sampling from the space of possible vocabulary sizes. This probabilistic segmentation allows for considering multiple ways of segmenting a sequence into sub-sequences. This idea can be widely used in different applications of protein informatics. In particular, we used it for (i) alignment-free discriminative protein sequence motif discovery method, called DiMotif, as well as (ii) variable-length extension of protein sequence embedding called ProtVecX.

We compared DiMotif against two existing tools for motif discovery: HH-Motif as an instance of non-discrminative methods, and DLocalMotif as an instance of discriminative methods. We compared the performances in the detection of 20 distinct sub-types of experimentally verified motifs. HH-Motif comparing HMMs of orthologs for retrieving SLiMs, achieved the best average F1 and the DiMotif with domain-specific segmentation achieved the second best F1. DiMotif achieved the highest recall, making it an ideal tool for finding a list of candidates for further experimental verification. In addition, we evaluated DiMotif by extracting motifs related to (i) integrins, (ii) integrin-binding proteins, and (iii) biofilm formation. We showed that the extracted motifs could reliably detect reserved sequences of the same phenotypes, as indicated by their high F1 scores. We also showed that DiMotif could detect experimentally verified motifs related to nuclear localization signals. By using KL divergence between the distribution of motifs in the positive sequences, DiMotif is capable of outputting multi-part motifs. DiMotif segmentation can be inferred once from Swiss-Prot dataset and then be used to extract the motif in a given discriminative motif mining problem setting. Unlike the existing alignment-based motif discovery methods, the input sequences to DiMotif do not need to be necessarily homologous sequences. Thus, it can be utilized in cases motifs need to be found from a set of non-homologous sequences.

Deep learning for protein secondary structure prediction

We studied the machine learning-based protein secondary structure prediction approaches from the protein primary sequence. We focused on finding an optimal representation and deep learning predictive model for this task, over the most challenging dataset for this task to-date, i.e., Q8 (8-way classification) on CullPDB/CB513 dataset, where the similar sequences to the training samples are removed from the test set.

We investigated (i) different protein sequence representations including one-hot vectors, biophysical features, protein sequence embedding (ProtVec), deep amino acid contextualized embedding (ELMo), and the Position Specific Scoring Matrix (PSSM), (ii) different deep-learning architectures including convolutional neural networks (CNN), recurrent neural networks (in particular Bi-LSTM), use of highway connection, attention mechanism, and multi-scale CNN (Jiyun Zhou et al. 2018). We showed that PSSM and its combination with one-hot vectors achieve the best performance in protein secondary structure prediction. The best performing model was the CNN-BiLSTM architecture, which captures both local and global sequence features essential for proteins secondary structure. Our tool, called *DeepPrime2Sec* provides the community with a specialized framework for the protein secondary structure prediction covering different architectures. The BiLSTM-CRF architecture performs competitive to the other existing approach in the literature, and the ensemble of the best performing model in Prime2Sec marginally outperforms the existing methods. Also, we performed error analysis on the most accurate model based on the location of misclassified amino acids as well as the confusion matrix analysis. Strikingly, misclassified secondary structures were significantly correlated with locating at the structural transitions. Such a correlation is most likely due to the inaccurate assignment of the secondary structure at the boundaries in ground-truth (Y. Yang et al. 2016). By ignoring the boundary amino acids from the evaluation, the Q8 accuracy would increase for an extra %20, i.e., %90.3. Analysis of the confusion matrix furthermore indicates that similar secondary structures are highly confusing (helices: H and G as well as unstructured regions: S, T, and L) showing that the model can learn high-level information about the secondary structures.

Chapter 3

Language-agnostic processing of genomics/metagenomics

3.1 Introduction and chapter overview

Microbial communities exist on every accessible surface on earth and have important functions relevant to supporting, regulating, and in some cases causing unwanted conditions (e.g., diseases) in their hosts/environments, ranging from organismal environments, such as the human body, to ecological environments, such as soil and water (R. Martin et al. 2014). These communities typically consist of a variety of microorganisms, including eukaryotes, archaea, bacteria, and viruses. Due to differences in nutrient availability and environmental conditions, microbial communities from different environments have widely varying taxonomic structures and compositions (Costello et al. 2009; Pinto et al. 2012; Moran 2015; Sunagawa et al. 2015; Fierer 2017; Eck et al. 2017; Duvallet et al. 2017).

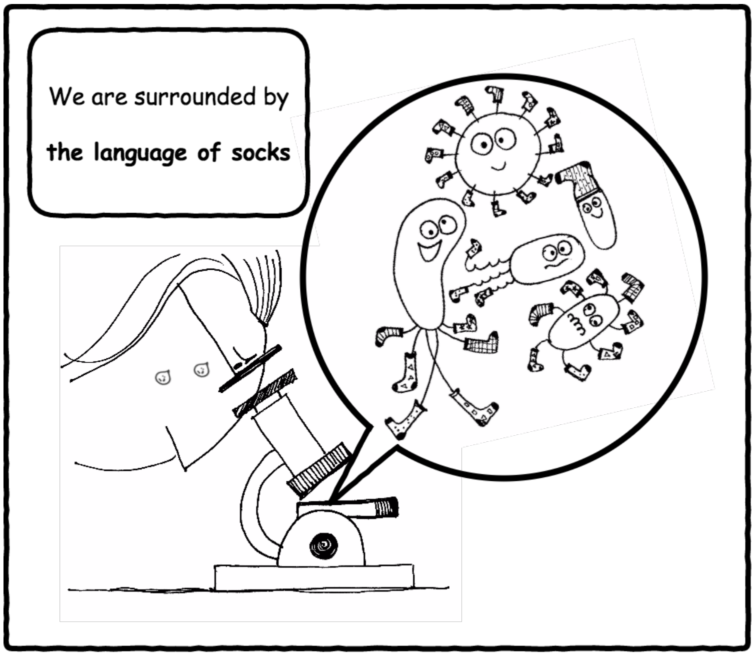
The human microbiota refers to all microorganisms living in close association with the human body. It is now widely believed that changes in our microbiota correlate with numerous diseases, raising the possibility that manipulation of these communities may be used to treat diseases. The microbiota (particularly the intestinal microbiota) is known to play important roles in healthy humans, including: (i) prevention of pathogen growth, (ii) education and regulation of the host immune system, and (iii) providing energy substrates to the host (Lynch and Pedersen 2016). Consequently, dysbiosis of the human microbiota

[¶] The content of this chapter is based on the following publications:

1. Asgari, E., Garakani, K., McHardy, A. C., & Mofrad, M. R. (2018). MicroPheno: Predicting environments and host phenotypes from 16S rRNA gene sequencing using a k-mer based representation of shallow sub-samples. *Bioinformatics*, 34(13), i32-i42, https://doi.org/10.1093/bioinformatics/bty296.

2. Asgari, E., Münch, P. C., Lesker, T. R., McHardy, A. C., & Mofrad, M. R. (2018). DiTaxa: Nucleotide-pair encoding of 16S rRNA for host phenotype and biomarker detection. *Bioinformatics*, bty954, https://doi.org/10.1093/bioinformatics/bty954.

can promote diseases, including asthma (Marsland et al. 2013; Arrieta et al. 2015), irritable bowel syndrome (Saulnier et al. 2011; I. Cho and Blaser 2012), Clostridium difficile infection (Cammarota et al. 2014), chronic periodontitis (Z. L. Deng et al. 2017; Jorth et al. 2014), cutaneous leishmaniasis (Gimblet et al. 2017), obesity (Turnbaugh et al. 2008; Ridaura et al. 2013), chronic kidney disease (Ramezani and Raj 2014), Ulcerative colitis (Michail et al. 2012), and Crohn's disease (Gevers et al. 2014; Pascal et al. 2017). For instance, the human microbiota appears to play a particularly important role in the development of Crohn's disease, an inflammatory bowel disease (IBD), with a prevalence of approximately 40 per 100,000 and 200 per 100,000 in children and adults, respectively (Kappelman et al. 2007). The role of human microbiota in human health motivates developing methods for inferring relationships between microbial taxa or functions associated with certain host phenotypes. Similarly, environmental microbial communities also serve important functions, such as nutrient cycling (Gilbert and Neufeld 2014). For instance, the microbiota living in the ocean account for half of the primary production on Earth (Moran 2015). The soil microbiome surrounding the root of plants impacts plant fertility and growth (Chaparro et al. 2012). Such studies are called *metagenomic studies*, as they deal with the collective genomes of microorganisms from environmental samples for inferring the microbial diversity and certain characteristics of the environments of interest. Metagenomics is a relatively new area of research in microbiology and is becoming increasingly important (R. Martin et al. 2014).



The starting point of many metagenomic studies is either 16S rRNA gene amplicon or shotgun metagenome sequencing of environmental samples (Pollock et al. 2018). 16S rRNA gene sequencing has several disadvantages in comparison with shotgun metagenomics, such as its inability to resolve functions, and accordingly functional variations within individual taxa (Poretsky et al. 2014; Ranjan et al. 2016; Cottier et al. 2018; Pollock et al. 2018). However, due to its low cost, 16S rRNA amplicon sequencing is still the most popular data type generated in microbiome studies (Hamady and R. Knight 2009; Pollock et al. 2018). The 16S rRNA gene is highly conserved across bacteria and archaea, includes both conserved regions, against which universal species-independent PCR primers can be directed, and nine hypervariable regions (V1-V9), which allow differential identification of taxon identities and relative abundances (Janda and Abbott 2007). After sequencing, the obtained data are usually processed with bioinformatics software such as QIIME (Caporaso et al. 2010; Lawley and Tannock 2017), Mothur (Schloss et al. 2009), or Usearch (Robert C. Edgar et al. 2011)] and clustered into groups of closely related sequences, referred to as Operational Taxonomic Units (OTUs).

Three main strategies for creating OTUs have been developed: in the *de novo* OTU clustering scheme, input sequences are aligned against one another and OTU clusters created based on a user-specified percent identity cutoff (in practice mostly 97%) without comparisons to reference databases. The implementation of the *de novo* strategy is difficult to parallelize and therefore limited to small-scale datasets. Variations of this method, such as sub-sample open-reference OTU picking (Rideout et al. 2014) or centroid-based greedy clustering approaches (W. Li and Godzik 2006) accelerate this process and enable their application to larger datasets. Alternatively, in closed-reference OTU clustering, input reads are aligned to a set of cluster centroids defined in a reference database (containing clusters of previously identified OTUs) and will be reported as an OTU, if they align at a given threshold. This strategy will not report OTUs for novel taxa that are not part of the reference database, though. An advantage is the usual high quality of taxonomic assignments of the reference database, which can be used for taxonomic assignment of the OTUs from the community of interest. Finally, the open-reference OTU clustering scheme combines *de novo* and closed-reference picking, where input sequences are aligned against a reference database (such as Greengenes (DeSantis et al. 2006)) and sequences that fail to match the reference are subsequently clustered *de novo* in a serial process (Rideout et al. 2014)). Individual algorithms for OTU clustering, post- and pre-processing have been combined to pipelines such as mothur (Schloss et al. 2009), QIIME (Caporaso et al. 2010; Lawley and Tannock 2017), USEARCH (Robert C. Edgar et al. 2011) and LotuS (Hildebrand et al. 2014).

Although OTU clustering has simplified 16S rRNA processing by substituting the analysis of millions of reads by analysis of only thousands of OTUs, it still has several disadvantages: OTUs do not necessarily represent meaningful taxonomic units, such as e.g. species, and sequencing errors may inflate diversity estimates by orders of magnitude (Kunin et al. 2010). To prevent diversity overestimates, OTU based approaches require a highly stringent quality control and relaxed clustering at $< 97\%$ similarity. While this approach limits the inflation of OTUs by potential sequencing errors, it comes at the expense of taxonomic resolution and may combine organisms with distinct biological properties and capabilities into a single OTU. A further disadvantage is that OTU calling requires extensive sequence alignment efforts. All of the above mentioned OTU-picking strategies involve sequence alignments either to the reference genomes or to the sample sequences, which is computationally expensive and cannot be easily extended to further samples. It was shown that OTUs were generally ecologically consistent across habitats, but observed OTU content can differ substantially between clustering methods (T. S. B. Schmidt et al. 2014). Since the number of obtained OTUs and their content is dependent on the pipeline and the parameter settings, reproducing the same analysis is difficult (Y. He et al. 2015). An alternative solution is the analysis of individual 16S rRNA gene sequence (Callahan et al. 2016; Amir et al. 2017; Nearing et al. 2018), which is computationally challenging, as each 16S rRNA sample may contain 10,000s of sequences. The main focus of this chapter is developing OTU-free methods for processing of 16S rRNA sequencing.

Chapter overview

We begin this chapter with a genomic phenotype prediction problem setting and show that language-agnostic k-mer representations of microbial genomes can be more effective than expensive genomic sequence events in the detection of the phenotype of interest. Next, we extend the setting to the metagenomics. In Section §3.3 In Section §3.3, we introduce MicroPheno, a reference- and alignment-free approach for predicting environments and host phenotypes from 16S rRNA gene sequencing based on sequence k-mer representations that benefit from a bootstrapping framework for investigating the sufficiency of shallow subsamples. Deep learning methods, as well as classical approaches, were explored for predicting environments and host phenotypes. MicroPheno is the state-of-the-art approach for the host phenotype prediction outperforming costly OTU features. Although MicroPheno could outperform OTU features in phenotype prediction, short k-mers cannot be easily used as taxa distinctive biomarkers and OTU features remained the state-of-the-art in biomarker detection (Segata, Izard, et al. 2011). In Section §3.4, we propose DiTaxa, an alignment- and reference- free, subsequence based paradigm for processing of 16S rRNA microbiome data for phenotype and biomarker detection. The main distinction of this approach from existing methods is substituting standard OTU-clustering (Robert C Edgar 2013) or sequence-level analysis (Callahan et al. 2016) by segmenting 16S rRNA reads into the most frequent variable-length subsequences of a dataset. we compare the performance of DiTaxa to the state-of-the-art methods using human-associated 16S rRNA samples for periodontal disease, rheumatoid arthritis, and inflammatory bowel diseases, as well as a synthetic benchmark dataset. We show that DiTaxa improved the state-of-the-art performance in biomarker detection over 16S rRNA data while performing competitively to the k-mer based state-of-the-art approach in phenotype prediction. Finally, in Section §3.5, we conclude with a summary of contributions this dissertation has had in metagenomics.

3.2 K-mer based representation for microbial genome phenotype prediction

Before facing the complexity of metagenomics, we investigate the strength of language-agnostic representations in the prediction of microbial phenotypes in a single genomics setting. In particular, we attempt to predict infection-causing strains of *Klebsiella pneumoniae*, which contains several loci that are related to virulence.

Genome representations

In a typical genotype-phenotype association scenario, the genomic DNAs of clinical microbial isolates of different phenotypes are sequenced into reads. Subsequently, the trimmed reads are assembled using genome assemblers, and then the assembled genomes are annotated using genome annotators. Next, the genes are clustered into gene families. Presence/absence of these genes (GPA) is one of the features used in the machine learning classification of genomes. The GPA features aside, single-nucleotide polymorphism (SNPs) are also extracted using variant calling pipelines and are the other features in the classification. Often the transcriptional profiles are also included as part of features (gene expression profiles). In fact, many of these processes are language-aware and much domain knowledge other than DNA/RNA sequences are needed (e.g. mappings to the references). Here, we would like to investigate whether sequence information alone would be sufficient for a machine learning-based prediction of the phenotype of interest. An alternative to the extraction of genomic events is to use directly sequence features. Frequency of overlapping k-mers in the sequences is a simple but effective representation for many bioinformatics tasks. Since for the detection of genomic events (e.g., GPA and SNPs) the domain knowledge is explicitly used (e.g., reference genomes or gene families) we consider them as language-dependent methods, while k-mer representation is completely language agnostic and use no other information than the genome sequence. Variety of predictive models are used for the machine learning-based prediction of microbial phenotypes, including support vector machine, random forests, deep neural networks, based on the problem setting and the number of training samples (Khaledi et al. 2019).

Method

Dataset

The dataset we use for this experiment is the *Klebsiella pneumoniae* dataset of sequencing reads belong to 178 isolates collected from six countries available at European Nucleotide Archive[*]. For these isolates, the binary clinical phenotype of human carriage status versus

[*] https://www.ebi.ac.uk/ena/data/view/PRJEB2111

human infection (including invasive infections) is provided. 142 *Klebsiella pneumoniae* isolates are associated with the'human infection' phenotype and 36 of them with 'human carriage'.

Representation Creation

Language-aware genomic events: The isolate reads obtained from European Nucleotide Archive, were assembled with SPAdes, v.3.0.1 (Bankevich et al. 2012). We annotate the assembled genomes via Prokka (v1.12) (Seemann 2014). Next, we cluster the gene sequences into gene families using Roary (Page et al. 2015). Subsequently, MAFFT is used to infer multiple sequence alignments (Katoh and Standley 2013). Next, MSA2VCF[*] for indels is utilized. Reads were mapped to the reference using Stampy (Lunter and Goodson 2011), followed by SNPs detection with SAMtools (H. Li et al. 2009). We distinguish between synonymous (not causing a change in the amino acid) versus non-synonymous SNPs (changing the amino acid).

K-mer representation: we generate overlapping k-mer frequency distributions from the sequenced isolates. the k value of 6 was chosen which is relatively small, in comparison with other k-mer representations in the literature (Aun et al. 2018). As discussed in §2.3, k-mer representations are popular in machine learning for all areas of bioinformatics research, including proteomics (Grabherr et al. 2011; Asgari and M. R. Mofrad 2015), genomics (Jolma et al. 2013; Alipanahi et al. 2015), epigenomics (Awazu 2016; Giancarlo et al. 2015), and metagenomics (Derrick E Wood and Steven L Salzberg 2014; Asgari, Garakani, et al. 2018). The reason behind introducing redundancy of overlapping k-mers is the uncertainty exists in the segmentation of the biological sequences.

Machine learning classifier

For the predictive modeling we utilize support vector machine (SVM) classifier (Cortes and Vapnik 1995) with a linear kernel, random forests (RF) (Cutler et al. 2012), and logistic regression (LR) (Cramer 2002). The size of the dataset is not large enough for proper training of neural network architectures. We use 80% of the data in cross-validation (cv) and the remaining 20% for test purpose. Two separate scenarios of splitting the test set and the cross-validation folds are defined: (i) a completely random splitting and (ii) a challenging setting where the cross-validation folds and the test set are all phylogenetically distant. We perform hyperparameter tuning of the classifiers optimizing for the F1-score in 10-fold cross-validation setting. The RF classifier are optimized for the macro F1-score over different hyperparameters, including (i) the number of decision trees in the ensemble, (ii) the number of features for computing the best node split, (iii) the function to measure the quality of a split and (iv) the minimum number of samples required to split a node. The logistic regression and the linear SVM are optimized for the macro F1-score over (i) the C parameter (inverse to the regularization strength) and (ii) class weights (to be uniform over all classes

[*] https://github.com/lindenb/jvarkit/

or proportional to the class frequencies). Subsequently, we evaluate the optimized classifier on the test set.

Results

We provide the results on the phylogenetically-informed split of cross-validation folds and the test set, separate from the random split. The classification results on prediction of bacterial phenotype to be 'human infection' versus 'human carriage' using random split is shown in Tables 3.2 and 3.2 respectively for the cross-validation and the test set. The classification performance on prediction of bacterial phenotype to be 'human infection' versus 'human carriage' using phylogenetically-aware split is shown in Tables 3.4 and 3.4 respectively for the cross-validation and the test set. The most important set of results for us would be the performance on the test set when the test set is phylogenetically distant from the training, i.e., Table 3.4. For selection of the best results we look at the maco-F1 score as it is the average of F1 for both positive and negative class and both classes are considered equally in this evaluation metric.

Interestingly, 6-mer, while being the least computationally expensive feature and language-agnostic, achieved the best macro-F1 of 84% and the F1 of 91%. The choice of classifier is highly dependent on the feature used. For the 6-mer SVM is the best performing classifier, while using GPA features, LR has been the best choice.

Conclusions

We showed that the language-agnostic features, i.e., k-mers can perform equal or even better than representation requiring domain knowledge and reference sequences to produce. In the case of human infection phenotype prediction for *Klebsiella pneumoniae* strains, the short k-mer (k=6) could achieve the best macro-F1 score in a challenging setting where both cross-validation folds and the test set were supposed to phylogenetically distant. The main disadvantage of short k-mers is that the most discriminative features cannot easily be mapped to functional units in comparison with GPA and SNPs, which are genomic events that are easier to track. In the next sections, we extend the k-mer representation for metagenomics setting (§3.3). Next, in §3.4, we provide a sub-sequence-based language-agnostic representation that having high predictive power aside, can also be mapped to certain taxonomic categories.

Table 3.1. Human infection phenotype prediction cross-validation results for *Klebsiella pneumoniae* strains

Feature	Classifier	Accuracy	Precision	Recall	F1	macro-F1
6-mer	LR	0.85	0.95	0.85	0.90	0.80
	RF	0.80	0.85	0.90	0.88	0.68
	SVM	0.87	0.96	0.87	0.91	0.83
GPA	LR	0.74	0.79	0.91	0.85	0.53
	RF	0.76	0.78	0.96	0.86	0.46
	SVM	0.72	0.80	0.85	0.82	0.55
GPA 6-mer	LR	0.73	0.80	0.87	0.83	0.56
	RF	0.81	0.85	0.92	0.88	0.68
	SVM	0.70	0.79	0.84	0.82	0.53
GPA non_syn_SNP	LR	0.74	0.83	0.83	0.83	0.62
	RF	0.77	0.79	0.97	0.87	0.49
	SVM	0.72	0.82	0.82	0.82	0.60
GPA syn_SNP	LR	0.76	0.80	0.92	0.86	0.56
	RF	0.77	0.78	0.97	0.87	0.46
	SVM	0.72	0.80	0.87	0.83	0.54
non_syn_SNP	LR	0.72	0.82	0.82	0.82	0.58
	RF	0.77	0.78	0.97	0.87	0.46
	SVM	0.69	0.78	0.83	0.81	0.51
non_syn_SNP 6-mer	LR	0.72	0.82	0.82	0.82	0.58
	RF	0.79	0.85	0.89	0.87	0.68
	SVM	0.72	0.81	0.84	0.82	0.56
non_syn_SNP syn_SNP	LR	0.78	0.78	0.99	0.88	0.47
	RF	0.77	0.78	0.98	0.87	0.43
	SVM	0.78	0.78	0.99	0.88	0.47
syn_SNP	LR	0.74	0.79	0.91	0.85	0.53
	RF	0.77	0.78	0.97	0.87	0.46
	SVM	0.76	0.79	0.94	0.86	0.52
syn_SNP 6-mer	LR	0.75	0.79	0.92	0.85	0.53
	RF	0.81	0.85	0.92	0.88	0.68
	SVM	0.76	0.79	0.94	0.86	0.52

Table 3.2. Human infection phenotype prediction test set results for *Klebsiella pneumoniae* strains

Feature	Classifier	Accuracy	Precision	Recall	F1	macro-F1
6-mer	LR	0.94	0.93	1.00	0.97	0.91
	RF	0.94	0.93	1.00	0.97	0.91
	SVM	0.94	0.93	1.00	0.97	0.91
GPA	LR	0.67	0.75	0.86	0.80	0.40
	RF	0.78	0.78	1.00	0.88	0.44
	SVM	0.67	0.75	0.86	0.80	0.40
GPA 6-mer	LR	0.67	0.75	0.86	0.80	0.40
	RF	0.94	0.93	1.00	0.97	0.91
	SVM	0.61	0.73	0.79	0.76	0.38
GPA non_syn_SNP	LR	0.67	0.75	0.86	0.80	0.40
	RF	0.78	0.78	1.00	0.88	0.44
	SVM	0.67	0.75	0.86	0.80	0.40
GPA syn_SNP	LR	0.72	0.80	0.86	0.83	0.56
	RF	0.78	0.78	1.00	0.88	0.44
	SVM	0.61	0.73	0.79	0.76	0.38
non_syn_SNP	LR	0.67	0.79	0.79	0.79	0.52
	RF	0.72	0.76	0.93	0.84	0.42
	SVM	0.72	0.80	0.86	0.83	0.56
non_syn_SNP 6-mer	LR	0.67	0.79	0.79	0.79	0.52
	RF	0.94	0.93	1.00	0.97	0.91
	SVM	0.67	0.75	0.86	0.80	0.40
non_syn_SNP syn_SNP	LR	0.67	0.75	0.86	0.80	0.40
	RF	0.78	0.78	1.00	0.88	0.44
	SVM	0.72	0.76	0.93	0.84	0.42
syn_SNP	LR	0.67	0.75	0.86	0.80	0.40
	RF	0.78	0.78	1.00	0.88	0.44
	SVM	0.67	0.75	0.86	0.80	0.40
syn_SNP 6-mer	LR	0.67	0.75	0.86	0.80	0.40
	RF	0.94	0.93	1.00	0.97	0.91
	SVM	0.67	0.75	0.86	0.80	0.40

Table 3.3. Human infection phenotype prediction phylogenetic-aware-cross-validation results for *Klebsiella pneumoniae* strains

Feature	Classifier	Accuracy	Precision	Recall	F1	macro-F1
6-mer	LR	0.87	0.97	0.86	0.91	0.84
	RF	0.83	0.90	0.86	0.88	0.77
	SVM	0.87	0.97	0.86	0.91	0.84
GPA	LR	0.78	0.86	0.85	0.85	0.70
	RF	0.76	0.76	0.98	0.86	0.43
	SVM	0.77	0.77	1.00	0.87	0.43
GPA 6-mer	LR	0.78	0.86	0.85	0.85	0.70
	RF	0.86	0.91	0.91	0.91	0.80
	SVM	0.77	0.77	1.00	0.87	0.43
GPA non_syn_SNP	LR	0.76	0.85	0.83	0.84	0.66
	RF	0.71	0.76	0.91	0.83	0.45
	SVM	0.76	0.84	0.85	0.84	0.65
GPA syn_SNP	LR	0.81	0.89	0.86	0.88	0.75
	RF	0.71	0.76	0.91	0.83	0.45
	SVM	0.79	0.91	0.80	0.85	0.74
non_syn_SNP	LR	0.78	0.90	0.80	0.85	0.72
	RF	0.71	0.77	0.89	0.83	0.48
	SVM	0.77	0.90	0.79	0.84	0.71
non_syn_SNP 6-mer	LR	0.78	0.90	0.80	0.85	0.72
	RF	0.87	0.92	0.91	0.92	0.82
	SVM	0.77	0.90	0.79	0.84	0.71
non_syn_SNP syn_SNP	LR	0.78	0.90	0.80	0.85	0.72
	RF	0.72	0.77	0.91	0.83	0.49
	SVM	0.76	0.89	0.77	0.83	0.70
syn_SNP	LR	0.76	0.89	0.77	0.83	0.70
	RF	0.71	0.76	0.91	0.83	0.45
	SVM	0.76	0.89	0.77	0.83	0.70
syn_SNP 6-mer	LR	0.76	0.89	0.77	0.83	0.70
	RF	0.87	0.90	0.94	0.92	0.81
	SVM	0.76	0.89	0.77	0.83	0.70

Table 3.4. Human infection phenotype prediction phylogenetically-distant test set results for *Klebsiella pneumoniae* strains

Feature	Classifier	Accuracy	Precision	Recall	F1	macro-F1
6-mer	LR	0.88	0.94	0.89	0.92	0.86
	RF	0.88	0.90	0.95	0.92	0.85
	SVM	0.90	0.94	0.92	0.93	0.88
GPA	LR	0.65	0.81	0.68	0.74	0.60
	RF	0.76	0.78	0.95	0.85	0.63
	SVM	0.69	0.71	0.95	0.81	0.41
GPA 6-mer	LR	0.65	0.81	0.68	0.74	0.60
	RF	0.86	0.89	0.92	0.91	0.82
	SVM	0.69	0.71	0.95	0.81	0.41
GPA non_syn_SNP	LR	0.65	0.83	0.65	0.73	0.61
	RF	0.67	0.73	0.86	0.79	0.49
	SVM	0.69	0.82	0.73	0.77	0.64
GPA syn_SNP	LR	0.67	0.83	0.68	0.75	0.63
	RF	0.67	0.73	0.86	0.79	0.49
	SVM	0.55	0.85	0.46	0.60	0.54
non_syn_SNP	LR	0.63	0.91	0.54	0.68	0.62
	RF	0.69	0.74	0.86	0.80	0.54
	SVM	0.63	0.91	0.54	0.68	0.62
non_syn_SNP 6-mer	LR	0.63	0.91	0.54	0.68	0.62
	RF	0.90	0.92	0.95	0.93	0.87
	SVM	0.63	0.91	0.54	0.68	0.62
non_syn_SNP syn_SNP	LR	0.63	0.95	0.51	0.67	0.62
	RF	0.67	0.73	0.86	0.79	0.49
	SVM	0.63	0.95	0.51	0.67	0.62
syn_SNP	LR	0.61	0.84	0.57	0.68	0.59
	RF	0.67	0.73	0.86	0.79	0.49
	SVM	0.61	0.84	0.57	0.68	0.59
syn_SNP 6-mer	LR	0.61	0.84	0.57	0.68	0.59
	RF	0.84	0.89	0.89	0.89	0.80
	SVM	0.61	0.84	0.57	0.68	0.59

3.3 K-mer based representation for predicting environments and host phenotypes

Microbial communities play important roles in the function and maintenance of various biosystems, ranging from the human body to the environment. A major challenge in microbiome research is the classification of microbial communities of different environments or host phenotypes. The most common and cost-effective approach for such studies to date is 16S rRNA gene sequencing. Recent falls in sequencing costs have increased the demand for simple, efficient, and accurate methods for rapid detection or diagnosis with proved applications in medicine, agriculture, and forensic science. We describe a reference- and alignment-free approach for predicting environments and host phenotypes from 16S rRNA gene sequencing based on k-mer representations that benefits from a bootstrapping framework for investigating the sufficiency of shallow sub-samples. Deep learning methods as well as classical approaches were explored for predicting environments and host phenotypes. K-mer distribution of shallow sub-samples outperformed Operational Taxonomic Unit (OTU) features in the tasks of body-site identification and Crohn's disease prediction. Aside from being more accurate, using k-mer features in shallow sub-samples allows (i) skipping computationally costly sequence alignments required in OTU-picking, and (ii) provided a proof of concept for the sufficiency of shallow and short-length 16S rRNA sequencing for phenotype prediction. In addition, k-mer features predicted representative 16S rRNA gene sequences of 18 ecological environments, and 5 organismal environments with high macro-F1 scores of 0.88 and 0.87. For large datasets, deep learning outperformed classical methods such as Random Forest and SVM.

Machine learning for 16S rRNA Gene Sequence Analysis

Popular main machine learning tasks over 16S rRNA gene sequencing data are taxonomic classification, host phenotype prediction, as well as biomarker detection. Several recent studies predicted the environment or host phenotypes using 16S gene sequencing data for body-sites (Knights et al. 2011; Statnikov et al. 2013), disease state (X. Xu et al. 2016; Eck et al. 2017; Duvallet et al. 2017), ecological environment quality status prediction (Cordier et al. 2017), and subject prediction for forensic science (Fierer et al. 2010; Schmedes et al. 2018). In all, OTUs served as the main input feature for the down stream machine learning algorithms. Random Forest and then, ranking second, linear Support Vector Machine (SVM) classifiers were reported as the most effective classification approaches in these studies (Statnikov et al. 2013; Pasolli et al. 2016; Carrieri et al. 2016; Duvallet et al. 2017).

Related prior work on body-site classification (Knights et al. 2011; Statnikov et al. 2013) used the following datasets: Costello Body Habitat (CBH - 6 classes), Costello Skin Sites (CSS - 12 classes) (Costello et al. 2009), and Pei Body Site (PBS - 4 classes) (Statnikov et al. 2013). An extensive comparison of classifiers for body-site classification over CBH, CSS, and PBS on top of OTU features has been performed by Statnikov et al (Statnikov et al. 2013). The best accuracy levels measured by relative classifier information (RCI) achieved by using

OTU features are reported as 0.784, 0.681, and 0.647 for CBH, CSS, and RCI respectively. Due to the insufficiency of the number of samples (on average 57 samples per class for CBH, CSS, and PBS) as well as the unavailability of raw sequences for some of the datasets mentioned above, instead of using the same dataset we replicate the state-of-the-art approach suggested in (Statnikov et al. 2013), i.e. Random Forest and SVM over OTU features for a larger dataset (Human Microbiome Project dataset). We then compare OTU features with k-mer representations. Working on a larger dataset allows for a more meaningful investigation and better training for deep learning approaches.

Detecting disease status based on 16S gene sequencing is becoming more and more popular, with applications in the prediction of Psoriasis (151 samples for 3 classes - Best accuracy: 0.225), IBD (patients: 49 samples, healthy:59 - Best AUC:0.95) (X. Xu et al. 2016), (patients: 91 samples, healthy: 58 samples - Best AUC:0.92) (Eck et al. 2017). Similar to body-site classification datasets, the datasets used for disease prediction were also relatively small. In this paper, we use the Crohn's disease dataset (Gevers et al. 2014) with 1359 samples (patients: 731 samples, negative class: 628 samples) for evaluating our proposed method and then compare it with the use of OTU features.

We focus on machine learning approaches for classification of environments or host phenotypes of 16S rRNA gene sequencing data, which is the most popular and cost-effective sequencing method for the characterization of microbiome to date (Pasolli et al. 2016; Pollock et al. 2018). Studies on the use of machine learning for predicting microbial phenotype instead of environments/host phenotype (Dutilh et al. 2013; Ross et al. 2013), as well as predictions based on shotgun metagenome and whole-genome microbial sequencing are beyond the scope of this paper, although we believe that one may easily adapt the proposed approach to shotgun metagenomics, similar to the study by Cui et al. on IBD prediction (Cui and Xuegong Zhang 2013).

Recently, deep learning methods became popular in various applications of machine learning in bioinformatics (Asgari and M. R. Mofrad 2015; Min et al. 2016) and in particular in microbiome research (Ditzler et al. 2015). However, to the best of our knowledge, this is the first study exploring environment and host phenotype prediction from 16S rRNA gene sequencing data with deep learning approaches.

16S rRNA gene sequence representations

OTU representation

As reviewed in §3.3, prior machine learning works on environment/host phenotype prediction have been mainly using OTU representations as the input features to the learning algorithm. Although there exist non-OTU based pipelines for 16S rRNA sequence analysis (e.g. DADA-2 (Callahan et al. 2016)), almost all popular 16S rRNA sequence processing pipelines cluster sequences into OTUs based on their sequence similarities, utilizing a variety of algorithms (N.-P. Nguyen et al. 2016; Lawley and Tannock 2017). QIIME allows OTU-picking using three different strategies: **(i) closed-reference OTU-picking:** sequences are compared against a marker gene database (e.g., Greengenes (D. McDonald et al. 2012) or SILVA (Quast et al. 2013)) to be clustered into OTUs and then the sequences different from the reference genomes beyond a certain sequence identity threshold are discarded. **(ii) open-reference OTU-picking:** the remaining sequences after a closed-reference calling go through a de novo clustering. This allows for using the whole sequences as well as capturing sequences belonging to new communities, which are absent in the reference databases (Rideout et al. 2014). **(iii) pure de novo OTU-picking:** sequences (or reads) are only compared among themselves and no reference database is used. The third strategy is more appropriate for novel species absent in the current reference. Although OTU clustering reduces the analysis of millions of reads to working with only thousands of OTUs and simplifies the subsequent phylogeny estimation and multiple sequence alignment, OTU representations have several shortcomings: **(i)** All three OTU-picking strategies involve massive amounts of sequence alignments either to the reference genomes (in closed/opened-reference strategies) or to the sequences present in the sample (in open-reference and de novo strategies) which makes them very expensive (Y. Cai et al. 2017) in comparison with reference-free/alignment-free representations. **(ii)** Overall sequence similarity is not a proper condition for grouping sequences and OTUs can be phylogenetically incoherent. For instance, a single mutation between two sequences is mostly ignored by OTU-picking algorithms. However, if the mutation does not occur within the sample, it might be a signal for assigning a new group. In addition, several mutations within a group most likely are not going to be tolerated by OTU-picking algorithms. However, having the same ratio across samples may suggest that the mutated sequences belong to the same group (Koeppel and M. Wu 2013; N.-P. Nguyen et al. 2016). **(iii)** The number of OTUs and even their contents are very sensitive to the pipeline and parameters, and this makes them difficult to reproduce (Y. He et al. 2015).

k-mer representations

k-mer count vectors have been shown to be suitable input features for performing machine learning on biological sequences for a variety of bioinformatics tasks (Marçais and Kingsford 2011). In particular, k-mer count features have been used for taxonomic classifications of microbial 16S and metagenome datasets (McHardy et al. 2007; Patil et al. 2011; Derrick E. Wood and Steven L. Salzberg 2014; Kawulok and Deorowicz 2015; Menzel and Krogh 2015; Vervier et al. 2016). However, to the best of our knowledge, k-mer features have not been explored for phenotypical and environmental characterizations of 16S rRNA sequence data.

In this paper we propose using k-mer representations of shallow sub-samples for predicting environments and host phenotypes from 16S rRNA sequences. Our approach is fast, reference-free and alignment-free, while contributing to building accurate classifiers outperforming conventional OTU features in body-site identification and Crohn's disease classification. We propose a bootstrapping framework to investigate the sufficiency of shallow sub-samples for the prediction of the phenotype of interest, which proves the sufficiency of short-length and shallow sequencing of 16S rRNA. In addition, we explore deep learning methods as well as classical approaches for classification. Furthermore, we demonstrate the value of

PCA, t-SNE, and supervised deep representation learning for visualization of microbial samples/sequences of different phenotypes. We also show that k-mer features can be used to predict representative 16S rRNA gene sequences from 18 ecological environments and 5 organismal environments with high macro-F1s.

Material and Methods

Datasets

Body-site identification

We employ the 16S rRNA gene sequence dataset provided by the NIH Human Microbiome Project (HMP) (Peterson et al. 2009; Huttenhower and Human Microbiome Project Consortium 2012)[*]. In particular, we use processed, annotated 16S rRNA gene sequences of up to 300 healthy individuals, each sampled at 4 major body-sites (oral, airways, gut, vagina) and up to three time points. For each major body-site, a number of sub-sites were sampled. We focus on 5 body sub-sites: anterior nares (nasal) with 295 samples, saliva (oral) with 299 samples, stool (gut) with 325 samples, posterior fornix (urogenital) with 136 samples, and mid vagina (urogenital) with 137 samples, in total 1192 samples. These body-sites are selected to represent differing levels of spatial and biological proximity to one another, based on relevance to pertinent human health conditions potentially influenced by the human microbiome. To compare k-mer based approach with state-of-the-art OTU features, we collect the closed-reference OTU representations of the same samples in HMP (Huttenhower and Human Microbiome Project Consortium 2012) [†] obtained using the QIIME pipeline (Rideout et al. 2014).

Crohn's disease prediction

For the classification of Crohn's disease, we use the 16S rRNA dataset described in (Gevers et al. 2014)[‡], which is currently the largest pediatric Crohn's disease dataset available. This dataset includes annotated 16S rRNA gene sequence data for 731 pediatric ($\leq$ 17 years old) patients with Crohn's disease and 628 samples verified as healthy or diagnosed with other diseases, making a total of 1359 samples. Sequencing here was targeted towards the V4 hypervariable region of the 16S rRNA gene. Similar to the body-site dataset, to compare the k-mer based approach with the approach based on OTU features, we collect the OTU representations of the same samples from Qiita repository [§] obtained using QIIME pipeline (Rideout et al. 2014).

[*] Available at http://hmpdacc.org/HM16STR/

[†] Available at https://qiita.ucsd.edu/study/descriptipn/1928

[‡] Available at: https://www.ncbi.nlm.nih.gov/bioproject/PRJEB13679

[§] Available at https://qiita.ucsd.edu/study/description/1939

Predicting the environment for representative 16S rRNA gene sequences

MetaMetaDB provides a comprehensive dataset of representative 16S rRNA sequences of various ecological and organismal environments, collected from existing 16S rRNA databases spanning almost 181 million raw sequences. In the MetaMetaDB pipeline, low-quality nucleotides, adapters, ambiguous sequences, homopolymers, duplicates, and reads shorter than 200bp, as well as chimeras have been removed and 16S rRNA sequences were clustered with 97% identity, generating 1,241,213 representative 16S rRNA sequences marked by their environment (C. C. Yang and Iwasaki 2014). MetaMetaDB divides its ecological environments into 34 categories and its organismal environments into 28 categories. We create three datasets, which are subsets of MetaMetaDB, to investigate the discriminative power of k-mers in predicting microbial habitats. Since the sequences in MetaMetaDB were already filtered and semi-identical sequences removed, OTU-picking would not be required, as it would result in an almost one-to-one mapping between the sequences and OTUs (we verified this using QIIME).

Ecological environment prediction: MetaMetaDB is imbalanced in terms of the number of representative sequences per environment. For this study, we pick the ecological environments with more than 10,000 samples, ending up wicorresponding toth 18 classes of ecological environments: activated sludge, ant fungus garden, aquatic, bioreactor, bioreactor sludge, compost, food, food fermentation, freshwater, freshwater sediment, groundwater, hot springs, hydrocarbon, marine, marine sediment, rhizosphere, sediment, and soil [*]. We make two datasets out of the sequences in these environments by stratified sampling: **ECO-18K** containing 1000 randomly selected instances per class (a total of 18K sequences) and **ECO-180K**, which is 10 times larger than **ECO-18K**, i.e. contains 10,000 randomly selected instances per class (a total of 180K sequences).

Organismal environment prediction: from the organismal environments in MetaMetaDB we select a subset containing gut microbiomes of 5 different organisms (bovine gut, chicken gut, human gut, mouse gut, termite gut) and down-sampled each class to the size of the smallest class by stratified sampling, resulting in 620 samples per class and a total of 3100 sequences. We call this dataset **5GUTS-3100**.

MicroPheno computational workflow

We describe using deep learning and classical methods for classification of the environments or host phenotypes of microbial communities using k-mer frequency representations obtained from shallow sub-sampling of 16S rRNA gene sequences. We propose a bootstrapping framework to confirm the sufficiency of using a small portion of the sequences within a 16S rRNA sample for determining the underlying phenotype. The MicroPheno computational workflow has the following steps (Figure 3.1): (i) to find the proper size N for the sample, such that it

[*] Datasets and descriptions are available at http://mmdb.aori.u-tokyo.ac.jp/download.html

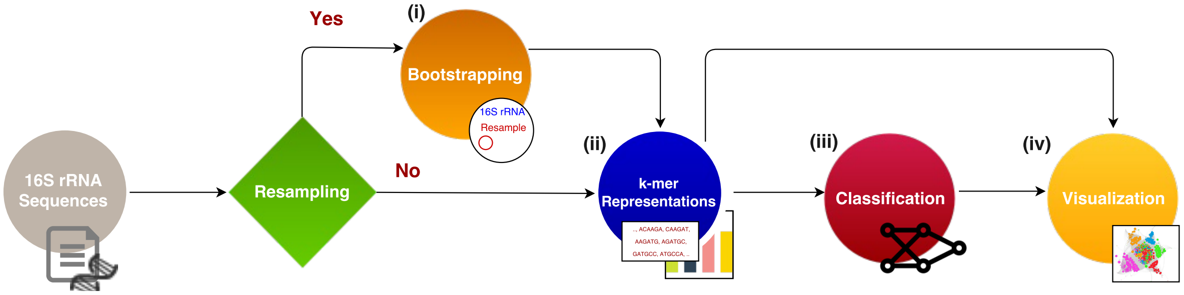


Figure 3.1. The components and the data flow in the MicroPheno computational workflow.

stays representative of the data and produces a stable k-mer profile, the 16S rRNA sequences go through a bootstrapping phase. (ii) Afterwards, the sub-sampled sequences are used to find the best value for k for classification, to produce the k-mer representations of the samples. (iii) The k-mer representations are used for classification with Deep Neural Networks (DNN), Random Forest (RF), and Linear SVM. (iv) Finally, the k-mer representations as well as the supervised representations trained using DNNs are used for visualization of the 16S rRNA gene sequences or samples. In what follows, these steps are explained in detail.

Bootstrapping: Confirming the sufficiency of only a small portion of 16S rRNA sequences for environment or host phenotype classification is important because (i) sub-sampling reduces the preprocessing run-time, and (ii) more importantly, it proves that even a shallow 16S rRNA sequencing is enough. We propose a resampling framework to give us quantitative measures for finding the proper sampling size. Let $\theta_k(X_i)$ be the normalized k-mer distribution of X_i, a set of sequences in the i^{th} 16S rRNA sample. We investigate whether only a portion of X_i, which we represent as $\widetilde{x}_{ij}$, i.e. j^{th} resample of X_i with sample size N, would be sufficient for producing a proper representation of X_i. To quantitatively find a sufficient sample size for X_i we propose the following criteria in a resampling scheme. **(i) Self-consistency**: resamples for a given size N from X_i produce consistent $\theta_k(\widetilde{x}_{ij})$'s, i.e. resamples should have similar representations. **(ii) Representativeness**: resamples for a given size N from X_i produce $\theta_k(\widetilde{x}_{ij})$'s similar to $\theta_k(X_i)$, i.e. similar to the case where all sequences are used.

We quantitatively define self-inconsistency and unrepresentativeness and seek parameter values that minimize them. We measure the **self-inconsistency** ($\bar{D_S}$) of the resamples' representations by calculating the average Kullback Leibler divergence among normalized k-mer distributions for N_R resamples (here N_R=10) with sequences of size N from the i^{th} 16S rRNA sample:

$$\bar{D_S}_i(N,k,N_R) = \frac{1}{N_R(N_R-1)} \sum_{\substack{\forall p,q \\ (p\neq q)}\in\{1,2,\cdots,N_R\}} D_{KL}(\theta_k(\widetilde{x}_{ip}),\theta_k(\widetilde{x}_{iq})),$$

where $|\widetilde{x_{il}}| = N; \forall l \in \{1,2,\cdots,N_R\}$. We calculate the average of the values of $\bar{D_S}_i(N,k,N_R)$

over the M different 16S rRNA samples:

$$\bar{D_S}(N,k,N_R) = \frac{1}{M}\sum_{i=1}^{M} \bar{D_{Si}}(N,k,N_R).$$

We measure the **unrepresentativeness ($\bar{D_R}$)** of the resamples by calculating the average Kullback Leibler divergence between normalized k-mer distributions for N_R resamples (N_R=10) with size N and using all the sequences in X_i for the i^{th} 16S rRNA sample:

$$\bar{D_{Ri}}(N,k,N_R) = \frac{1}{N_R}\sum_{\forall p\in\{1,2,\cdots,N_R\}} D_{KL}(\theta_k(\widetilde{x}_{ip}),\theta_k(X_i)),$$

where $|\widetilde{x_{il}}| = N$; $\forall l \in \{1,2,\cdots,N_R\}$. We calculate the average over $\bar{D_{Ri}}(N,k)$'s for the M 16S rRNA samples:

$$\bar{D_R}(N,k,N_R) = \frac{1}{M}\sum_{i=1}^{M} \bar{D_{Ri}}(N,k,N_R).$$

For the experiments on body-site and the dataset for Crohn's disease, we measure self-inconsistency $\bar{D_S}$ and unrepresentativeness $\bar{D_R}$ for $N_R = 10$ and $M = 10$ for any $8 \geq k \geq 3$ with sampling sizes ranging from 20 to 10000. Each point in Figure 3.4 represents the average of 100 ($M \times N_R$) resamples belonging to M randomly selected 16S rRNA samples, each of which is resampled $N_R = 10$ times. Since in the ecological and organismal datasets each sample is a single sequence, the bootstrapping step is skipped.

k-mer representation: We propose using the $l1$ normalized k-mer distribution of 16S rRNA sequences as input features for machine learning classification algorithms as well as visualization. Normalizing the representation allows for having a consistent representation, even when the sampling size is changed. For each k-value we pick a sampling size that gives us a self-consistent and representative representation measured by $\bar{D_S}(N,k,N_R)$ and $\bar{D_R}(N,k,N_R)$, respectively, as explained above.

Classification: Random Forests and linear SVM are the state-of-the-art classical approaches for categorical prediction on 16S rRNA sequences (Statnikov et al. 2013; Pasolli et al. 2016; Duvallet et al. 2017) and in general for many machine learning problems in bioinformatics (Olson et al. 2017). These two approaches, which are respectively instances of non-linear and linear classifiers, are both adopted in this study. In addition to these classical approaches, we also evaluate the performance of deep Neural Network classifiers in predicting environments and host phenotypes.

We evaluate and tune the model parameter in a stratified 10 fold cross-validation scheme. To ensure optimizing for both precision and recall, we optimize the classifiers for the harmonic mean of precision and recall, i.e. F1. In particular, to give equal importance to the classification categories, specifically when we have imbalanced classes, we use macro-F1, which is the average of F1's over categories. Finally the evaluation metrics are averaged over the folds and the standard deviation is also reported. We provide both micro and macro metrics which are averaged over instances and over categories, respectively.

Classical learning algorithms: We use a one-versus-rest strategy for multi-class linear SVM (Suykens and Vandewalle 1999) and tune parameter C, the penalty term for regularization. Random Forest (Breiman 2001) classifiers are tuned for (i) the number of decision trees in the ensemble, (ii) the number of features for computing the best node split, and (iii) the function to measure the quality of a split.

Deep learning: We use the Multi-Layer-Perceptrons (MLP) Neural Network architecture with several hidden layers using Rectified Linear Unit (ReLU) as the nonlinear activation function. We use the softmax activation function at the last layer to produce the probability vector that can be regarded as representing posterior probabilities (Goodfellow et al. 2016). To avoid overfitting, we perform early stopping and also use dropout at hidden layers (N. Srivastava et al. 2014b). A schematic visualization of our Neural Networks is depicted in Figure 3.2. Our objective is minimizing the loss, i.e. cross entropy between output and the one-hot vector representation of the target class. The error (the distance between the output and the target) is used to update the network parameters via a Back-propagation algorithm using Adaptive Moment Estimation (Adam) as the optimizer (Kingma and Ba 2015). We start with a single hidden layer and incrementally increase the number of layers with systematic exploration of the number of hidden units and dropout rates to find a proper architecture. We stop adding layers when increasing the number of layers does not result in achieving a higher macro-F1 anymore. In addition, for the visualization of samples we use the output of the $(n-1)^{th}$ hidden layer. Later in the results DNN-nL is a short form for a MLP Neural Network with n layers.

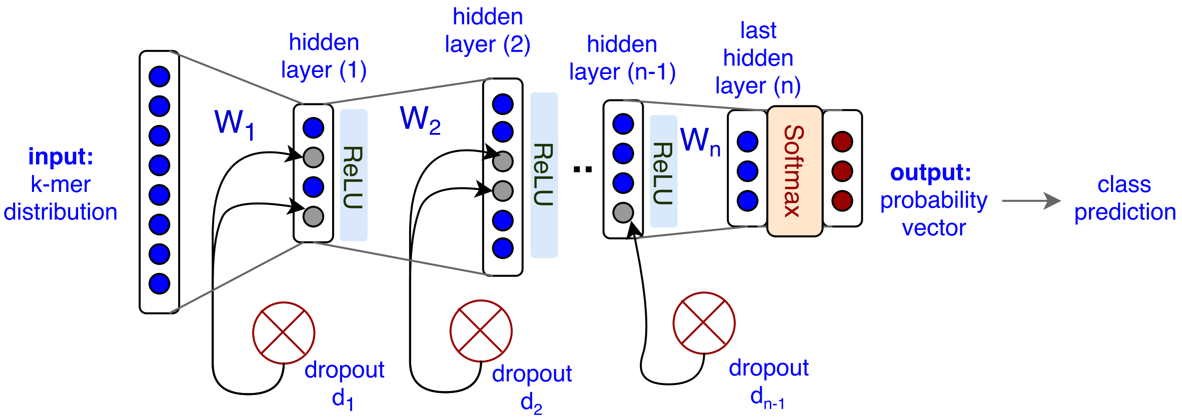


Figure 3.2. General architecture of the MLP Neural Networks that have been used in this study for multi-class classification of environment and host phenotypes.

Visualization: To project 16S rRNA sequencing samples to 2D for visualization purposes, we explore Principal Component Analysis (PCA) (Jolliffe and Jolliffe 1986) as well as t-

Distributed Stochastic Neighbor Embedding (t-SNE) (Van Der Maaten and G E Hinton 2008), as instances of respectively linear and non-linear dimensionality reduction methods. In addition, we explore the use of supervised deep representation learning in visualization of data (Bengio et al. 2013), i.e. we visualize the activation function of the last hidden layer of the Neural Network trained for prediction of environments or host phenotypes to be compared with unsupervised methods. The visualizations helps in obtaining a better understanding on how samples are distributed in a high dimensional space and how neural networks can obtain a transformation that separates different categories. More details on visualization methods are provided as supplementary materials.

Implementations: MicroPheno uses implementations of Random Forest, SVM, t-SNE, and PCA in the Python library scikit-learn (Pedregosa and Varoquaux 2011) and Deep Neural Networks are implemented in the Keras* deep learning framework using the TensorFlow back-end.

Results

In this section, the results are organized based on datasets. As discussed in Section 3.3, we have several choices in each step in the computational workflow: choosing the value of k in k-mer, the sampling rate, and the classifiers. To explore the parameter space more systematically, we followed the steps demonstrated in Figure 3.3. (i) In the first step, for each value of $8 \geq k \geq 3$, we pick a stable sample size based on the output of bootstrapping. (ii) We perform the classification task using tuned Random Forest for different k values and their selected sampling sizes based on boostrapping. We selected Random Forest, because we found it easy to tune and because it oftentimes outperforms linear SVM (Statnikov et al. 2013; Olson et al. 2017). (iii) As the third step, for a selected k, we investigate the role of sampling size (N) in classification. (iv) Finally, we compare different classifiers for the selected k and N. We also compare the performance of our proposed k-mer features with that of OTU features in classification tasks.

Body-site identification and Crohn's disease prediction

(i) Bootstrapping for sampling rate selection for k-mers: Higher k values require higher sampling rates to produce self-consistent and representative representations (Figure 3.4 for body-site dataset). As the structure of the curve for Crohn's disease dataset is similar to the body-site dataset, to avoid redundancy, the figure for Crohn's disease is provided as supplementary material. For each k, we consider a certain threshold on $\bar{D_S}$ and $\bar{D_R}$, to ensure selecting a sampling size resulting in self-consistent and representative representations.

(ii) Classification for different values of k with a sampling size selected based on the output of bootstrapping: Interestingly, using only 3-mer features with a very low

*https://keras.io/

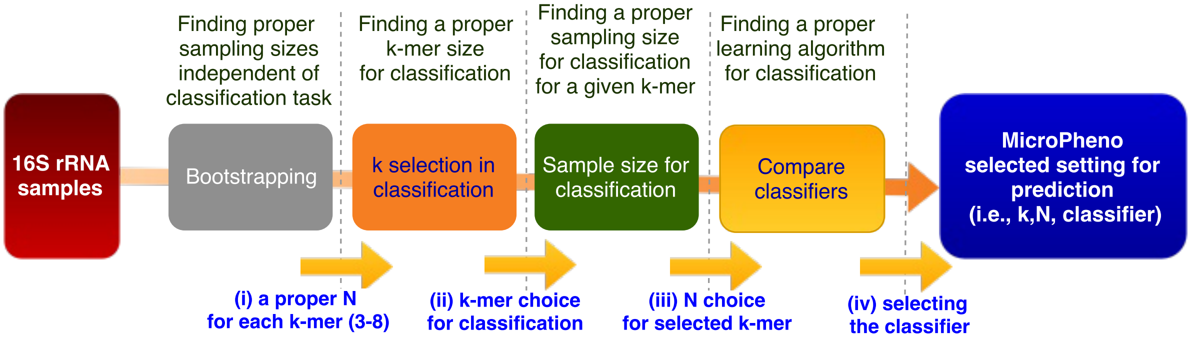

Finding proper sampling sizes independent of classification task
Finding a proper k-mer size for classification
Finding a proper sampling size for classification for a given k-mer
Finding a proper learning algorithm for classification
16S rRNA samples
Bootstrapping
k selection in classification
Sample size for classification
Compare classifiers
MicroPheno selected setting for prediction (i.e., k,N, classifier)
(i) a proper N for each k-mer (3-8)
(ii) k-mer choice for classification
(iii) N choice for selected k-mer
(iv) selecting the classifier

Figure 3.3. Steps we take to explore parameters for the representations and how we choose the classifier for prediction of the phenotype of interest in this study

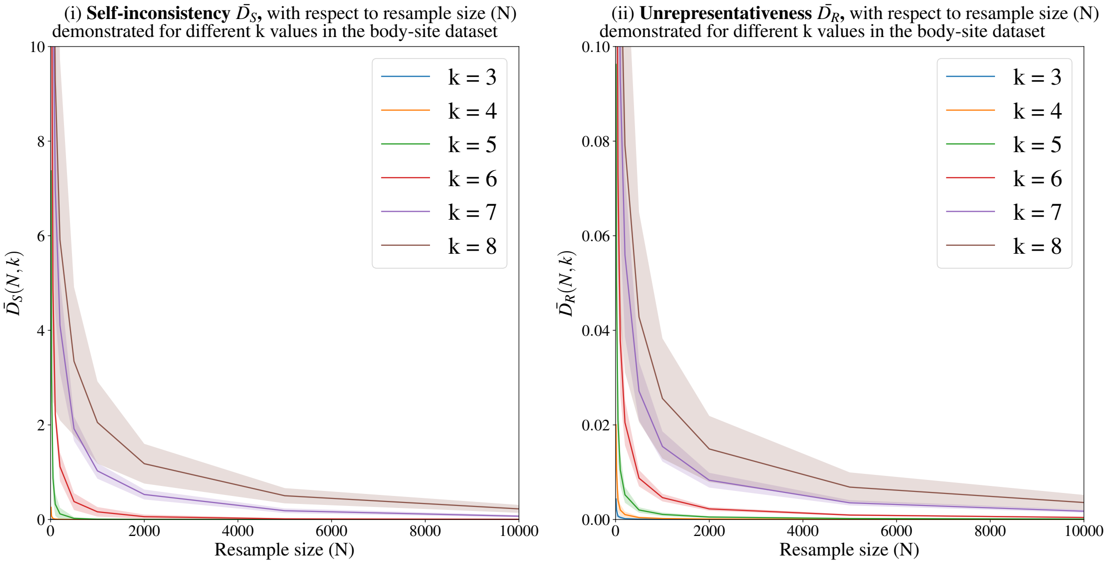


Figure 3.4. Measuring (i) self-inconsistency ($\bar{D_S}$), and (ii) unrepresentativeness ($\bar{D_R}$) for the body-site dataset. Each point represents an average of 100 resamples belonging to 10 randomly selected 16S rRNA samples. Higher k values require higher sampling rates to produce self-consistent and representative samples.

sampling rate (≈20/15000=0.0013) provides a relatively high performance for 5-way body-site classification (Table 3.5). The value of macro-F1 increases with the value of k from 3 to 6, but increasing k further than that does not have any additional effect on macro-F1 (Table 3.5, body-site dataset, step (ii)). For Crohn's disease Choosing k=6 with a sampling size of 2000 (≈2,000/38,000 = 0.05) provided a macro-F1 of 0.75 which is the minimum k with top performance (Table 3.5, Crohn's disease dataset, step (ii)).

(iii) Exploring the sampling size (N) for a selected k-mer: For a selected k-value (k=6), using the Random Forest classifier for different sampling sizes is presented in Table 3.5, step (iii) for the body-site and Crohn's disease datasets. **Body-site classification:** the

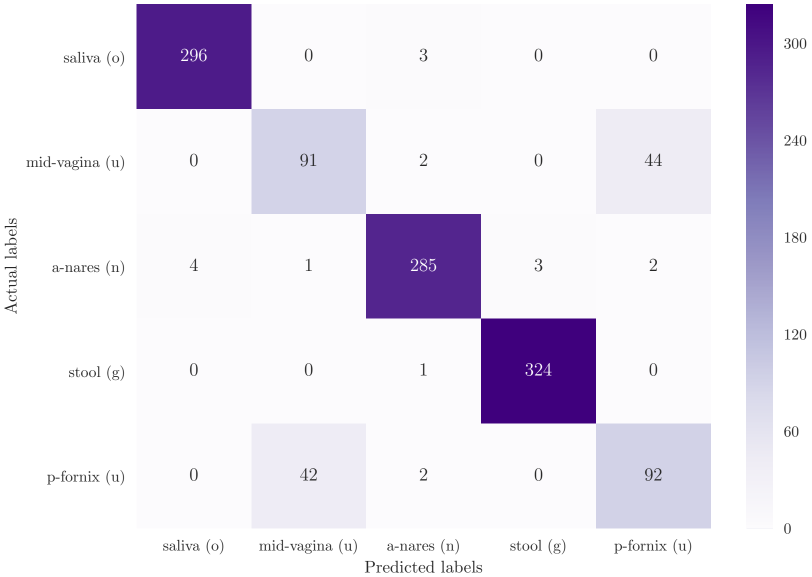


Figure 3.5. The confusion matrix for the classification of 5 major body-sites, using Random Forest classifier in a 10xfold cross-validation scheme. The presented body-sites are saliva (**o:** oral), mid-vagina (**u:** urogenital), anterior nares (**n:** nasal), stool (**g:** gut), and posterior fornix (**u:** urogenital).

results suggest that changing the sampling size from 0.6% to 100% of the sequences will not change the classification results substantially, suggesting that in body-site identification, a very shallow sub-sampling of the sequences is sufficient for a reliable prediction. Using more sequences does not necessarily increase the discriminative power and may even result in over-fitting. We selected a sampling size of 5000 for 6-mers (the sampling size with the highest macro-F1 and the minimum standard deviation) for comparison between classifiers in the next step. **Crohn's disease dataset:** increasing the sampling size from 100 (100/38000=0.003) to 5000 (5000/38000=0.13) increased the macro-F1 from 0.7 to 0.75. However, using all sequences instead of 0.13 of them in each sample, did not increase the discriminative power (Table 3.5).

(iv) Comparison of classifiers for the selected N, k: In the body-site prediction task, the Random Forest classifier obtained the top macro-F1 (0.84) for this 5-way classification (Table 3.5, step (iv)). The confusion matrix in Figure 3.5 shows that the most difficult decision for the classifier is to distinguish between mid vagina and posterior fornix, both of which are urogenital body-sites. As shown in the last row of Table 3.5, combining the urogenital body-sites increases the macro-F1 to 0.99 ± 0.01 using the Neural Network. Similarly for the Crohn's disease prediction dataset, the Random Forest classifier obtained the top macro-F1 (0.75) for this binary classification (Table 3.5, step (iv)).

The visualizations of body-site as well as Crohn's disease samples obtained through using

Table 3.5. The results for classification of major body-sites as well as Crohn's disease prediction using k-mer representations. The set of rows matches the steps (ii to iv) mentioned in Figure 3.3, i.e k-mer selection, N (sample size) selection, and finally selection of the classifier. The classifiers (Random Forest, Support Vector Machine and Neural Network classifiers) are tuned and evaluated in a stratified 10xfold cross-validation setting. The last row shows the Neural Network's performance in the classification of body-sites when the urogenital body-sites are combined.

Dataset	Step	Representation	Resample size	Classifier	Micro-metrics (averaged over samples)			Macro-metrics (averaged over classes)		
					Precision	Recall	F1	Precision	Recall	F1
Body-site (≈15000 reads/sample)	(ii)	3-mers	20	RF	0.84 ± 0.02	0.84 ± 0.02	0.84 ± 0.02	0.75 ± 0.03	0.75 ± 0.03	0.74 ± 0.03
		4-mers	100		0.86 ± 0.03	0.86 ± 0.03	0.86 ± 0.03	0.77 ± 0.03	0.77 ± 0.03	0.77 ± 0.03
		5-mers	500		0.89 ± 0.02	0.89 ± 0.02	0.89 ± 0.02	0.82 ± 0.03	0.82 ± 0.03	0.82 ± 0.03
		6-mers	2000		**0.91 ± 0.03**	**0.91 ± 0.03**	**0.91 ± 0.03**	**0.85 ± 0.05**	**0.85 ± 0.04**	0.84 ± 0.05
		7-mers	5000		**0.91 ± 0.03**	**0.91 ± 0.03**	**0.91 ± 0.03**	**0.85 ± 0.05**	**0.85 ± 0.05**	**0.85 ± 0.05**
		8-mers	8000		0.9 ± 0.03	0.9 ± 0.03	0.9 ± 0.03	0.85 ± 0.05	0.84 ± 0.05	0.84 ± 0.05
Crohn's disease (≈38000 reads/sample)	(ii)	3-mers	20	RF	0.62 ± 0.05	0.62 ± 0.05	0.62 ± 0.05	0.62 ± 0.05	0.61 ± 0.05	0.61 ± 0.05
		4-mers	100		0.7 ± 0.05	0.7 ± 0.05	0.7 ± 0.05	0.69 ± 0.05	0.69 ± 0.05	0.69 ± 0.05
		5-mers	500		0.74 ± 0.05	0.74 ± 0.05	0.74 ± 0.05	0.74 ± 0.05	0.74 ± 0.05	0.74 ± 0.05
		6-mers	2000		**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**
		7-mers	5000		**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**
		8-mers	8000		**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**
Body-site	(iii)	6-mers	100	RF	0.89 ± 0.02	0.89 ± 0.02	0.89 ± 0.02	0.82 ± 0.04	0.82 ± 0.03	0.81 ± 0.03
			1000		0.9 ± 0.03	0.9 ± 0.03	0.9 ± 0.03	0.83 ± 0.04	0.83 ± 0.04	0.83 ± 0.04
			2000		**0.91 ± 0.03**	**0.91 ± 0.03**	**0.91 ± 0.03**	**0.85 ± 0.05**	**0.85 ± 0.04**	**0.84 ± 0.05**
			5000		0.9 ± 0.03	0.9 ± 0.03	0.9 ± 0.03	**0.85 ± 0.04**	0.84 ± 0.04	**0.84 ± 0.04**
			10000		0.9 ± 0.03	0.9 ± 0.03	0.9 ± 0.03	0.84 ± 0.05	0.84 ± 0.05	0.84 ± 0.05
			All sequences		0.9 ± 0.03	0.9 ± 0.03	0.9 ± 0.03	0.84 ± 0.05	0.84 ± 0.04	0.84 ± 0.05
Crohn's disease	(iii)	6-mers	100	RF	0.71 ± 0.04	0.71 ± 0.04	0.71 ± 0.04	0.71 ± 0.04	0.7 ± 0.04	0.7 ± 0.04
			1000		0.75 ± 0.04	0.75 ± 0.04	0.75 ± 0.04	0.76 ± 0.04	0.75 ± 0.04	0.75 ± 0.04
			2000		0.75 ± 0.04	0.75 ± 0.04	0.75 ± 0.04	0.75 ± 0.04	0.75 ± 0.04	0.75 ± 0.04
			5000		**0.76 ± 0.04**	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**
			10000		**0.76 ± 0.04**	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.76 ± 0.05**	**0.75 ± 0.04**	**0.75 ± 0.05**
			All sequences		**0.76 ± 0.04**	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.76 ± 0.05**	**0.75 ± 0.04**	**0.75 ± 0.05**
Body-site	(iv)	6-mers	5000	RF	**0.9 ± 0.03**	**0.9 ± 0.03**	**0.9 ± 0.03**	**0.85 ± 0.04**	**0.84 ± 0.04**	**0.84 ± 0.04**
				SVM	0.86 ± 0.02	0.86 ± 0.02	0.86 ± 0.02	0.76 ± 0.06	0.76 ± 0.03	0.74 ± 0.04
				DNN-5L	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.79 ± 0.02	0.79 ± 0.03	0.79 ± 0.02
Crohn's disease	(iv)	6-mers	5000	RF	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**
				SVM	0.68 ± 0.04	0.68 ± 0.04	0.68 ± 0.04	0.68 ± 0.04	0.67 ± 0.04	0.67 ± 0.04
				DNN-7L	0.7 ± 0.02	0.7 ± 0.02	0.7 ± 0.02	0.7 ± 0.03	0.7 ± 0.02	0.7 ± 0.03
Body-site	-	6-mers	5000	DNN-4L (4 classes)	**0.99 ± 0.01**	**0.99 ± 0.01**	**0.99 ± 0.01**	**0.99 ± 0.01**	**0.99 ± 0.01**	**0.99 ± 0.01**

Table 3.6. Comparison of k-mers and OTU features in body-site classification as well as the detection of the Crohn's disease phenotype. For this comparison, Random Forest classifier (as an instance of non-linear classifiers) and linear SVM (as an instance of linear classifiers) have been used. The classifiers are tuned and evaluated in a stratified 10xfold cross-validation setting.

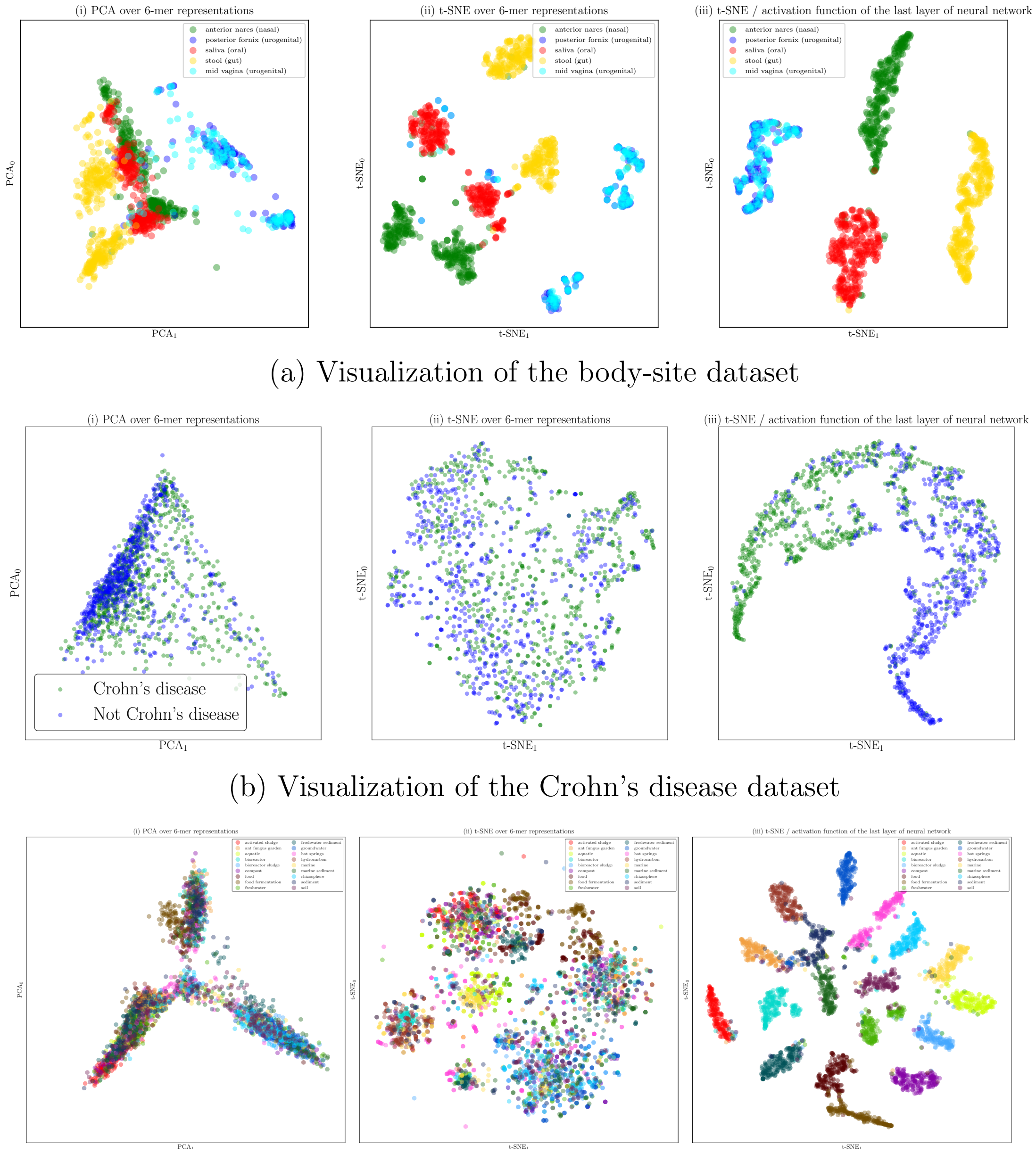
Dataset	Features	Classifiers	Micro-metrics (averaged over samples)			Macro-metrics (averaged over classes)		
			Precision	Recall	F1	Precision	Recall	F1
Body-site	6-mer features (size: 4096)	RF	**0.9 ± 0.03**	**0.9 ± 0.03**	**0.9 ± 0.03**	**0.85 ± 0.04**	**0.84 ± 0.04**	**0.84 ± 0.04**
		SVM	0.86 ± 0.02	0.86 ± 0.02	0.86 ± 0.02	0.76 ± 0.06	0.76 ± 0.03	0.74 ± 0.04
	OTU features (size: 20589)	RF	0.89 ± 0.02	0.89 ± 0.02	0.89 ± 0.02	0.83 ± 0.03	0.83 ± 0.03	0.83 ± 0.03
		SVM	0.85 ± 0.03	0.85 ± 0.03	0.85 ± 0.03	0.77 ± 0.05	0.78 ± 0.04	0.76 ± 0.04
Crohn's disease	6-mer features (size: 4096)	RF	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.75 ± 0.04**	**0.75 ± 0.04**
		SVM	0.68 ± 0.04	0.68 ± 0.04	0.68 ± 0.04	0.68 ± 0.04	0.67 ± 0.04	0.67 ± 0.04
	OTU features (size: 9511)	RF	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04
		SVM	0.68 ± 0.04	0.68 ± 0.04	0.68 ± 0.04	0.68 ± 0.04	0.68 ± 0.04	0.68 ± 0.04

Table 3.7. The results for the task of selecting between 18 ecological environments as well as 5 organismal environments belonging to 5 organisms' gut. The classifiers (Random Forest, Support Vector Machine and Neural Network classifiers) are tuned and evaluated in a stratified 10xfold cross-validation setting in three datasets ECO-18K, 5GUTS-3100, and ECO-180K. The step column refers to the steps in Figure 3.3.

Step	Representation	Dataset	Classifier	Micro-metrics (averaged over samples)			Macro-metrics (averaged over classes)		
				Precision	Recall	F1	Precision	Recall	F1
(ii)	3-mers	ECO-18K	RF	0.6 ± 0.01	0.6 ± 0.01	0.6 ± 0.01	0.63 ± 0.02	0.6 ± 0.01	0.57 ± 0.01
	4-mers			0.67 ± 0.01	0.67 ± 0.01	0.67 ± 0.01	0.7 ± 0.01	0.67 ± 0.01	0.65 ± 0.01
	5-mers			0.72 ± 0.01	0.72 ± 0.01	0.72 ± 0.01	0.74 ± 0.01	0.72 ± 0.01	0.71 ± 0.01
	6-mers			**0.75 ± 0.01**	**0.75 ± 0.01**	**0.75 ± 0.01**	**0.76 ± 0.01**	**0.75 ± 0.01**	**0.73 ± 0.01**
	7-mers			0.74 ± 0.01	0.74 ± 0.01	0.74 ± 0.01	**0.76 ± 0.01**	0.74 ± 0.01	**0.73 ± 0.01**
	8-mers			0.72 ± 0.01	0.72 ± 0.01	0.72 ± 0.01	0.74 ± 0.01	0.72 ± 0.01	0.71 ± 0.01
(ii)	3-mers	5GUTS-3100	RF	0.8 ± 0.02	0.8 ± 0.02	0.8 ± 0.02	0.8 ± 0.02	0.8 ± 0.02	0.79 ± 0.02
	4-mers			0.84 ± 0.01	0.84 ± 0.01	0.84 ± 0.01	0.84 ± 0.01	0.84 ± 0.01	0.83 ± 0.01
	5-mers			0.86 ± 0.02	0.86 ± 0.02	0.86 ± 0.02	0.86 ± 0.02	0.86 ± 0.02	0.85 ± 0.02
	6-mers			**0.87 ± 0.01**	**0.87 ± 0.01**	**0.87 ± 0.01**	0.87 ± 0.01	**0.87 ± 0.01**	**0.87 ± 0.01**
	7-mers			**0.87 ± 0.01**	**0.87 ± 0.01**	**0.87 ± 0.01**	**0.88 ± 0.02**	**0.87 ± 0.01**	**0.87 ± 0.01**
	8-mers			0.86 ± 0.01	0.86 ± 0.01	0.86 ± 0.01	0.87 ± 0.01	0.86 ± 0.01	0.86 ± 0.01
(iv)	6-mers	ECO-18K	RF	0.75 ± 0.01	0.75 ± 0.01	0.75 ± 0.01	0.76 ± 0.01	0.75 ± 0.01	0.73 ± 0.01
			SVM	**0.79 ± 0.01**	**0.79 ± 0.01**	**0.79 ± 0.01**	**0.79 ± 0.01**	**0.79 ± 0.01**	**0.79 ± 0.01**
			DNN-3L	0.78 ± 0.01	0.78 ± 0.01	0.78 ± 0.01	0.78 ± 0.01	0.78 ± 0.01	0.78 ± 0.01
(iv)	6-mers	5GUTS-3100	RF	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01
			SVM	**0.88 ± 0.02**	**0.88 ± 0.02**	**0.88 ± 0.02**	**0.89 ± 0.01**	**0.88 ± 0.02**	**0.88 ± 0.02**
			DNN-5L	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.01
(iv)	6-mers	ECO-180K (10x larger)	RF	0.83 ± 0.0	0.83 ± 0.0	0.83 ± 0.0	0.84 ± 0.0	0.83 ± 0.0	0.83 ± 0.0
			SVM	0.86 ± 0.0	0.86 ± 0.0	0.86 ± 0.0	0.87 ± 0.01	0.86 ± 0.0	0.86 ± 0.0
			DNN-5L	**0.88 ± 0.0**	**0.88 ± 0.0**	**0.88 ± 0.0**	**0.88 ± 0.0**	**0.88 ± 0.0**	**0.88 ± 0.0**

PCA, t-SNE over raw k-mer representations, and t-SNE on the activation function of the last layer of the trained Neural Networks are presented in Figure 3.6 (a,b). These results suggest that supervised training of representations using Neural Networks provides a non-linear transformation of data that can discriminate between dissimilar environments and host phenotypes.

Comparison of k-mer and OTU features in prediction: For a comparison between OTU features and k-mer representations in body-site identification and Crohn's disease prediction, the Random Forest classifier (as an instance of non-linear classifier) and linear SVM (as an instance of linear classifier) were tuned and evaluated in a stratified 10xfold cross-validation setting. Our results suggest that for both k-mer features and OTUs, Random Forest is the best choice (Table 3.6). In addition, with almost $\frac{1}{5}$ (body-site dataset) and $\frac{1}{2}$ (Crohn's disease dataset) of the size of OTU features and in spite of being considerably less expensive to calculate, k-mers marginally outperforms OTU features in both body-site identification and Crohn's disease classification.



(a) Visualization of the body-site dataset

(b) Visualization of the Crohn's disease dataset

(c) Visualization of 18 ecological microbial environments

Figure 3.6. Visualization of (a) body-site, (b) Crohn's disease, (c) ecological environments datasets using different projection methods: (i) PCA over 6-mer distributions with unsupervised training, (ii) t-SNE over 6-mer distributions with unsupervised training, (iii) visualization of the activation function of the last layer of the trained Neural Network (projected to 2D using t-SNE).

Ecological and organismal environment prediction

(ii) Classification for different values of k: As stated before, for the ecological and organismal datasets we do not need to perform resampling, as we classify single, representative 16S rRNA sequences for the environment of interest. We thus can skip steps (i) and (iii) (Figure 3.3). Step (ii) in Table 3.7 shows the effect of k in the performance of the classification of the 18 ecological environments for the ECO-18K dataset and 5 organismal environments for 5GUTS-3100. **Ecological environments:** using k=6 provides the best classification performance, with a macro-F1 of 0.73 which is relatively high for a 18-way classification (has a mere 0.06 chance of randomly being assigned correctly for balanced dataset). **Organismal environments:** the results show that k=6 and 7 provide a high classification macro-F1 of 0.87 for 5 classes (0.2 chance of randomly occurring).

(iv) Comparison of classifiers for the selected k: For selected values of k, the results of the environment prediction with the Random Forest, SVM, and Neural Network classifiers are provided (Table 3.7, step (iv), ECO-18K and 5GUTS-3100 datasets). The SVM classifier obtained the top macro-F1s of 0.79 and 0.88 respectively for 18-way and 5-way classifications.

To see the effect of increasing the number of data points in classification performance we repeat the classifier comparison (step iv) for the ECO-180K dataset. The results are summarized in Table 3.7-ECO-18K dataset, showing that feeding more training instances results in better training for the deep learning approach, which outperformed the SVM and achieved a macro-F1 of 0.88, which is very high for a 18-way classification task. In training the Neural Networks for the ECO-18K dataset, increasing the number of hidden layers from 3 to more did not help result in improvements. However, using the ECO-180K dataset, which is 10 times larger, allowed us to train a deeper network and increased the macro-F1 by 5 percent going from 3 layers to 5 layers. Increasing the number of layers further did not result in any improvements.

Neural Network Visualization: Visualizations of representative 16S rRNA gene sequences in 18 ecological environments obtained through using PCA, t-SNE, and t-SNE on the activation function of the last layer of the trained Neural Network are presented in Figure 3.6 (c). For ease of visualization, we randomly picked 100 samples per class. These results suggest that supervised training of representations using Neural Networks provides a non-linear transformation of data containing information about high-level similarities between environments in the sub-plot on the right (scatter plot (iii)), where such structures appeared in the visualization only when more hidden layers were used: **(i)** On the left, the environments containing water are clustered in a dense neighborhood: marine, aquatic, freshwater, hot springs, bioreactor sludge (described in the source: “Bioreactor sludge is usually the sludge inside the bioreactor that treats waste-water.”), groundwater, and, surprisingly, rhizosphere (an environment where plants, soil, water, microorganisms, and nutrients meet and interact). **(ii)** In the middle, environments labeled related to sediment are found: sediment, freshwater sediment, marine sediment, and soil. **(iii)** Environments containing food, like food, food fermentation, and compost are at the bottom of the plot.

Discussion and Conclusion

In this work, MicroPheno, a new approach for predicting environments and host phenotypes on 16S rRNA gene sequencing has been presented, which uses k-mer representations of shallow sub-samples. We conclude with discussing the results of this study in three parts: (1) the use of k-mers versus OTUs, (2) the benefits of shallow sub-sampling, and (3) classical methods versus the deep learning approach.

K-mers versus OTUs: To evaluate MicroPheno, we compared k-mer representations with OTU features in tasks of body-site identification and Crohn's disease classification. We replicated the state-of-the-art approach, i.e. Random Forest over OTU features, on datasets larger than those that were previously explored. We showed that k-mer features outperform conventional OTUs, while having several advantages over OTUs: **(i)** The k-mer representations are easy to compute at no computational cost for any type of alignment to references or tasks of finding pair-wise sequence similarity within samples as in OTU-picking pipelines. Just to get an idea of the computational efficiency of k-mer calculation in comparison with OTUs, note that 6-mer distribution calculations have been ≈13 times and ≈20 faster than going through the OTU picking pipelines respectively for the Crohn's disease dataset and the Human Microbiome Project dataset; using the same number of threads. More details are provided in the supplementary material. **(ii)** Taxonomy-independent analysis is often the preferred approach for amplicon sequencing when the samples contain unknown taxa. k-mer features can be used without making assumptions about the taxonomy. However, OTU-picking pipelines make assumptions about the taxonomy as discussed in §3.3; therefore they can even be phylogenetically incoherent. **(iii)** The k-mer distribution is a well-defined representation, while OTUs are sensitive to the pipeline and the parameters. **(iv)** Sequence similarities are naturally incorporated in the k-mer representations for the downstream learning algorithm, while with grouping sequences into certain categories, sequence similarities between OTUs are ignored.

The main disadvantage of k-mer features over OTUs is that using short k-mers makes it more difficult to trace the relevant taxa to the phenotype of interest. When such an analysis is needed, using OTUs or increasing the size of k would be an alternative solution. However, as long as prediction is concerned, using a k-mer representation seems to be the best choice for an accurate and rapid detection/diagnosis over 16S rRNA sequencing samples.

Shallow sub-sampling: We proposed a bootstrapping framework to investigate the consistency and representativeness of k-mer distributions for different sampling rates. Our results suggest that, depending on the k-mer size, even very low sub-sampling rates (0.001 to 0.1, for k between 3 to 7) not only can provide a consistent representation, but can also result in better predictions while possibly avoiding overfitting. Setting aside the saving in preprocessing time as a natural benefit of sampling, this result also suggests that at least for similar phenotypes of interest, shallow sequencing of the microbial community is sufficient for accurate prediction.

Classical classifiers versus deep learning: We studied the use of deep learning methods as well as classic machine learning approaches for distinguishing among human body-sites, diagnosis of Crohn's disease, and predicting the environments from representative 16S gene sequences. Studying the role of dataset size in the classification of ecological environments showed that for large datasets (in our experiments 10K samples per class) using deep learning provides us with more accurate predictions. However, when the number of samples is not large enough, Random Forests performed better on both OTUs and k-mer features. In addition, we observed that for classification over representative sequences as opposed to samples (pool of sequences), the SVM outperformed the Random Forest classifier. Another advantage of using deep learning in classification was that supervised training of a proper representation of data results in a more discriminative representation for the downstream visualization compared to the unsupervised methods (PCA and t-SNE on the raw k-mer distributions). For body-site identification and even more clearly in ecological environment classification, the model was able to extract more high-level similarities between the environments.

3.4 Nucleotide-pair encoding for biomarker and phenotype detection

Identifying combinations of taxa distinctive for microbiome-associated diseases is considered key to the establishment of diagnosis and therapy options in precision medicine and imposes high demands on accuracy of microbiome analysis techniques. We propose subsequence based 16S rRNA data analysis, as a new paradigm for microbiome phenotype classification and biomarker detection. This method and software called DiTaxa substitutes standard OTU-clustering or sequence-level analysis by segmenting 16S rRNA reads into the most frequent variable-length subsequences. These subsequences are then used as data representation for downstream phenotype prediction, biomarker detection and taxonomic analysis. Our proposed sequence segmentation called nucleotide-pair encoding (NPE) is an unsupervised data-driven segmentation inspired by Byte-pair encoding, a data compression algorithm. The identified subsequences represent commonly occurring sequence portions, which we found to be distinctive for taxa at varying evolutionary distances and highly informative for predicting host phenotypes. We compared the performance of DiTaxa to the state-of-the-art methods in disease phenotype prediction and biomarker detection, using human-associated 16S rRNA samples for periodontal disease, rheumatoid arthritis and inflammatory bowel diseases, as well as a synthetic benchmark dataset. DiTaxa identified 17 out of 29 taxa with confirmed links to periodontitis (recall= 0.59), relative to 3 out of 29 taxa (recall= 0.10) by the state-of-the-art method. On synthetic benchmark data, DiTaxa obtained full precision and recall in biomarker detection, compared to 0.91 and 0.90, respectively. In addition, machine-learning classifiers trained to predict host disease phenotypes based on the NPE representation performed competitively to the state-of-the art using OTUs or k-mers. For the rheumatoid arthritis dataset, DiTaxa substantially outperformed OTU features with a macro-F1 score of 0.76 compared to 0.65. Due to the alignment- and reference free nature, DiTaxa can efficiently run on large datasets. The full analysis of a large 16S rRNA dataset of 1359 samples required ≈1.5 hours on 20 cores, while the standard pipeline needed ≈6.5 hours in the same setting.

16S rRNA Gene Sequence Representations for Biomarker Detection

Popular machine learning tasks over 16S rRNA gene sequencing data are taxonomic classification, host phenotype prediction, as well as biomarker detection. Although k-mer features and some other non-OTU features have been also used (Asgari, Garakani, et al. 2018; Callahan et al. 2016), the most common representation of 16S rRNA gene sequences is based on OTUs. Random Forest was reported as the most effective classification approach for several diseases (Statnikov et al. 2013; Carrieri et al. 2016; Duvallet et al. 2017; Eck et al. 2017; Asgari, Garakani, et al. 2018). Recently, we have shown that using k-mer representations of shallow-subsamples is computationally inexpensive (being reference- and alignment-free) and marginally outperforms OTUs in host phenotype and environment classification tasks (Asgari, Garakani, et al. 2018). However, a disadvantage of k-mer features is that short k-mers cannot easily be mapped to a taxonomy to obtain taxonomic biomarkers. Microbiome studies often aim to identify OTUs, taxa, or clades that differ in their abundance across two or more subsets of the input samples (e.g. between diseased and healthy states), here referred as biomarker discovery (Kuczynski et al. 2010; Donovan H Parks and Beiko 2012). Identifying these biological informative taxa that are enriched in only a subset of phenotypes (e.g. diseased subjects or patients that better respond to a certain treatment) is a challenging task, in particular for metagenomic samples, because of their high-dimensionality, sequencing errors, as well as other systematic biases, such as the presence of chimeric sequences (Kunin et al. 2010). One prominent biomarker example is the over-representation of the Firmicutes phylum in obese individuals compared to lean controls (R. E. Ley et al. 2005; R. Ley et al. 2006; Turnbaugh et al. 2008). In case the over-representations are causal for the aetiology of the diseases, detection of such biomarkers might have potential therapeutic implications if disease progression can be reversed by targeting over-expressed causative species using emerging technologies such as CRISPR/Cas9 or phage-based targeting (Mimee et al. 2015; Sheth et al. 2016; Fuente-Nuñez and Lu 2017). This is also true for biomarkers that are inversely related to disease progression, such as the over-representation of *Akkermansia muciniphila* in individuals with a healthier metabolic status and better clinical outcome after caloric restriction (Dao et al. 2016). For many other diseases where a microbiological component is expected, such (combinations) of biomarkers yet have to be found. Even when the biomarkers fail to be causal, they may enable prediction of the disease state or disease sub-types, or suggest suitable therapies in personalized medical interventions.

Different methods have been developed to identify OTU-based biomarkers (Paulson et al. 2013). The most widely used method is linear discriminant analysis effect size (LEfSe), which has a particular focus on high-dimensional class comparison for metagenomic analysis and determines features (such as taxa, OTUs, genes or clades) most likely to explain differences between two or more classes from relative OTU abundances (Segata, Izard, et al. 2011). This method uses the non-parametric factorial Kruskal-Wallis (KW) sum rank test (Kruskal and Wallis 1952). Several other with similar functionality exist that use different statistical tests over sample profiles based on OTU features, such as STAMP (Donovan H. Parks et al. 2014), MetaStats (Segata, Izard, et al. 2011) and MetagenomeSeq (Paulson et al. 2013)).

In this paper, we propose DiTaxa, an alignment- and reference- free, subsequence based paradigm for processing of 16S rRNA microbiome data for phenotype and biomarker detection. DiTaxa substitutes standard OTU-clustering by segmenting 16S rRNA sequences into variable length subsequences. The obtained subsequences are then used as data representation for downstream phenotype and biomarker detection. We show that DiTaxa outperforms the state-of-the-art approach in biomarker detection for synthetic and a number of disease-related datasets. In addition, DiTaxa performs competitively with the k-mer based state-of-the-art approach, outperforming OTU-features, in phenotype prediction.

Material and Methods

Datasets

Inflammatory Bowel Diseases

We use the largest pediatric Crohn's disease dataset available to date, described in (Gevers et al. 2014)[*], which covers different types of Inflammatory Bowel Diseases (IBD) (The same as Crohn's disease dataset presented in Section §3.3). This is a dataset of 1359 labeled 16S rRNA samples from 731 pediatric ($\leq$ 17 years old) patients diagnosed with Crohn's disease (CD), 219 with ulcerative colitis (UC), 73 with indeterminate colitis (IC), and 336 samples verified as healthy. Sequencing was targeted towards the V4 hypervariable region of the 16S rRNA gene. We downloaded OTU representations of the samples from Qiita repository [†] obtained using QIIME pipeline (Rideout et al. 2014).

Rheumatoid arthritis

We downloaded read data (454 platform) of the 16S rRNA gene sequences of V1 and V2 rRNA for 114 fecal DNA samples of a rheumatoid arthritis (RA) study (Scher, Sczesnak, et al. 2013) from SRA (ID: SRP023463). OTU clustering was performed based on filtered reads ($365.7k, 23.0\%$) of which 140,382 were unique and 119,217 singletons and resulted in 949 OTUs based on 97% identity. The OTU clustering pipeline is detailed in §3.4.

Periodontal disease

We use the data provided by Jorth, et *al.* (Jorth et al. 2014) to differentiate between healthy and diseased periodontal microbiota[‡]. This dataset consists of microbial samples collected from subgingival plaques from 10 healthy and 10 patients diagnosed with periodontitis. Sequencing was targeted towards the ($V4 - V5$) hypervariable region of the 16S rRNA gene. Similar to the RA dataset, we obtain the OTU features using the clustering pipeline detailed in §3.4.

Synthetic dataset

To evaluate DiTaxa in a known setting, we generated a dataset with synthetic samples using Grinder v. 0.5.3 (Angly et al. 2012) based on 1000 V4 regions of different genera of Green-genes (GG) sequences (DeSantis et al. 2006). V4 regions were extracted from the Green-genes 13.8 databases using the forward and reverse primer sequences GTGCCAGC[AC]GCCGCGGTAA and ATTAGA[AT]ACCC[CGT][AGT]GTAGTCC. To generate 16S rRNA datasets, the $length_bias$ parameter was set to zero and the *unidirectional* parameter was set to one. To cover the full V4 region, the amplicon read length distribution was set to 300 and the fold

[*] Available at: https://www.ncbi.nlm.nih.gov/bioproject/PRJEB13679

[†] Available at https://qiita.ucsd.edu/study/description/1939

[‡] Downloaded from http://datadryad.org/resource/doi:10.5061/dryad.d41v4

coverage of the input reference sequence was set to 30. We specified the percent of reads in the amplicon libraries that should be chimeric sequences to 10%. We used default parameters for the specification of the chimera distribution resulting in 89% bimeras, 11% trimeras and 0.3% quadmeras. Sequencing errors were introduced in the reads, at positions that follow a uniform model using the default ratio of substitutions to the number of indels (4 substitutions for each indel). Two sets of samples were created, denoted as *case* and *control* samples with an average number of sequences in both groups of 29,204 reads. While in the control set all 1000 genera were set to the same abundance (mean abundance set to 0.1%, 500 randomly selected GG V4 sequences (corresponding to unique genera) were enriched at equal levels in the case dataset (μ of non-selected genera set to 0.05%; μ of selected genera set to 0.15;). For both, the control and the case settings, 100 samples were generated, each with variations under the normal distribution ($\sigma = 0.02$). We processed the synthetic dataset using a standard pipeline consisting of USEARCH and UPARSE and generated 1,041 OTUs at 97% identity similarity, detailed in §3.4.

Nucleotide-pair Encoding

The idea of Nucleotide-pair Encoding (NPE) is inspired by the Byte Pair Encoding (BPE) algorithm, a simple universal text compression scheme (Gage 1994; Shibata et al. 1999), which has been also used for compressed pattern matching in genomics (L. Chen et al. 2004). Although BPE had lost its popularity for a long time in compression, only recently it again became popular, but for a different reason, i.e. word segmentation in machine translation in natural language processing (NLP). BPE became a common approach for a data-driven unsupervised segmentation of words into their frequent subwords, which facilitate open vocabulary neural network machine translation and improve the quality of translation by reducing the vocabulary size (Sennrich et al. 2016; Kudo 2018). In this work, we adapt the BPE algorithm for splitting biological sequences into frequent variable length subsequences called Nucleotide-pair Encoding (NPE). We propose NPE as general purpose segmentation for the biological sequence (DNA, RNA, and proteins). In contrast to the use of BPE in NLP for vocabulary size reduction, we use this method to increase the size of symbols from 4 nucleotides to a large set of variable length biomarkers.

The input to NPE is a set of sequences. We treat each sequence as a list of characters (nucleotides in the case of 16S rRNA gene sequences). The algorithm finds the most frequently occurring pair of adjacent symbols in the sequences. On the next, we replace all instances of the selected pair with a new subsequence (merged pair as a new symbol). The algorithm repeats this process until reaching a certain vocabulary size or when no more frequently occurring pairs of symbols available. The obtained merging operations can be inferred once from a large set of sequences in an offline manner and then applied to an unseen set of sequences. A simple pseudo-code of NPE is provided in Algorithm 2.

```
Data: $Seqs$ = Set of 16S sequences from all samples, $V$ = vocabulary size
Result: $S$ = Divided sequences into variable subsequences, $Merge\_opt$ = merging
        operations
$Sym = \{A, T, C, G\}$;
$S$ = list of $Seqs$, where each sequence is list of symbols $\in Sym$;
$Merge\_opt = stack()$;
$SymbFreq$ = mapping symbol pairs in $S$ to their frequencies;
while $|Sym| < V$ do
    sym1, sym2 = argmax ($SymbFreq$);
    $S$ = merge all consecutive sym1 & sym2 into $< sym1, sym2 >$ in $S$;
    $Sym.push(< sym1, sym2 >)$;
    $Merge\_opt.push(sym1, sym2)$;
    update($SymbFreq$) ; // For efficiency purpose we only update the symbol
     pairs that overlap with occurrences of the affected pairs
end
```

Algorithm 2: Adapted Byte-pair algorithm (BPE) for segmentation of DNA sequences (NPE)

Standard 16S rRNA gene processing workflow

To evaluate the performance of DiTaxa against the state-of-the-art, we used a standard 16S rRNA gene processing workflow employed in previous studies on 16S rRNA data (Taft et al. 2014; Lin Wang et al. 2016; Bindels et al. 2016). Note that throughout this paper "the standard pipeline (STDP)" refers to this workflow:

1. Obtained 16S rRNA gene sequencing reads are quality controlled and clustered using the Usearch 8.1 software package [*], where quality filtering is done with $fastq\_filter(-fastq\_maxee1)$.

2. The OTU clusters and representative sequences are determined using the UPARSE algorithm ($derep\_fulllength : minuniquesize2; cluster\_otus : otu\_radius\_pct3$) (Robert C Edgar 2013).

3. The next step is taxonomy assignment using the EZtaxon database (Yoon et al. 2017) as the reference database, and the decision is made by RDP Classifier (Q. Wang et al. 2007).

4. The OTU absolute abundance table and mapping file are used for statistical analyses in LDA Effect Size (LEfSe) (Segata, Izard, et al. 2011).

[*] http://www.drive5.com/usearch/

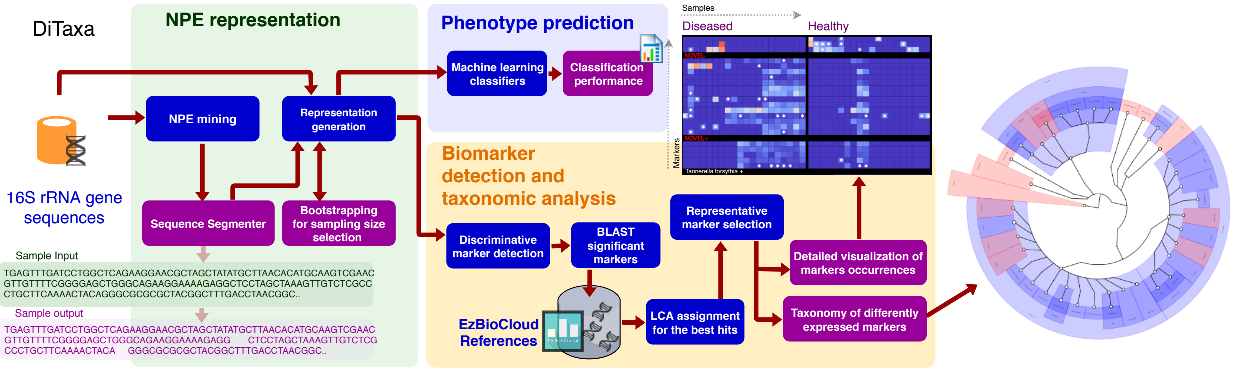

DiTaxa
NPE representation
16S rRNA gene sequences
NPE mining
Representation generation
Sequence Segmenter
Bootstrapping for sampling size selection
Sample Input
TGAGTTTGATCCTGGCTCAGAAGGAACGCTAGCTATATGCTTAACACATGCAAGTCGAAC
GTTGTTTTCGGGGAGCTGGGCAGAAGGAAAAGAGGCTCCTAGCTAAAGTTGTCTCGCC
CTGCTTCAAAACTACAGGGCGCGCGCTACGGCTTTGACCTAACGGC..
Sample output
TGAGTTTGATCCTGGCTCAGAAGGAACGCTAGCTATATGCTTAACACATGCAAGTCGAAC
GTTGTTTTCGGGGAGCTGGGCAGAAGGAAAAGAGG CTCCTAGCTAAAGTTGTCTCG
CCCTGCTTCAAAACTACA GGGCGCGCGCTACGGCTTTGACCTAACGGC..
Phenotype prediction
Machine learning classifiers
Classification performance
Biomarker detection and taxonomic analysis
Discriminative marker detection
BLAST significant markers
Representative marker selection
EzBioCloud References
LCA assignment for the best hits
Detailed visualization of markers occurrences
Taxonomy of differently expressed markers
Samples
Diseased
Healthy
NOVEL-
NOVEL+
Markers
Tannerella forsythia +

Figure 3.7. Computational workflow of DiTaxa, DiTaxa has three main components: (i) NPE representation, (ii) Phenotype prediction, (ii) Biomarker detection and taxonomic analysis. The purple boxes denote the outputs of the approach.

DiTaxa computational workflow

The DiTaxa computational workflow has three main components; (i) NPE representation creation, (ii) phenotype prediction, and (iii) biomarker detection and taxonomic analysis (shown in Figure 3.7). In this section, these components are described in details.

NPE representation

The first component of DiTaxa is the NPE representation creation. The 16S rRNA gene sequences aggregated from all samples from all phenotypes go through the NPE algorithm 2 for training segmentation operations. Then the segmentation will be applied on sequences to segment sequences into variable length subsequences. We pick the vocabulary size large enough to obtain discriminative 16S rRNA subsequences considered as biomarkers. Each sample will be presented as a count distribution of its subsequences. We propose a bootstrapping scheme to investigate the sufficiency of shallow sub-samples to produce proper representation.

In a previous study, using a bootstrapping framework we showed that shallow sub-samples of 16S rRNA gene sequences are sufficient to produce a proper k-mer presentation of data for phenotype prediction (Asgari, Garakani, et al. 2018). Similarly, here we use bootstrapping to investigate sufficiency and consistency of NPE representation, when only a small portion of the sequences are used. This has two important implications, first, sub-sampling reduces the preprocessing run-time, second, it shows that even a shallow 16S rRNA sequencing is enough for the phenotype prediction. We use a resampling framework to find a proper sampling size. Let $\theta_{\#npe}(X_i)$ be the normalized NPE (with vocabulary size of $\#npe$) distribution of X_i, a set of sequences in the i^{th} 16S rRNA sample. We investigate whether only a portion of X_i, which we represent as $\widetilde{x}_{ij}$, i.e. j^{th} resample of X_i with sample size N, would be sufficient for producing a proper representation of X_i. To find a sufficient sample size for X_i quantitatively, we propose the following formulation in a resampling scheme. **(i) Self-consistency**: resamples for a given size N from X_i produce consistent $\theta_{\#npe}(\widetilde{x}_{ij})$'s, i.e. resamples should have similar representations.(ii) Representativeness: resamples for a given size N from X_i produce $\theta_{\#npe}(\widetilde{x}_{ij})$'s similar to $\theta_{\#npe}(X_i)$, i.e. similar to the case where all sequences are used. As presented in (Asgari, Garakani, et al. 2018), we measure the **self-inconsistency ($\bar{D_S}$)** of the resamples' representations by calculating the average Kullback Leibler divergence among normalized NPE distributions for N_R resamples (here N_R=10) with sequences of size N from the i^{th} 16S rRNA sample:

$$\bar{D_S}_i(N, \#npe, N_R) = \frac{1}{N_R(N_R-1)} \sum_{\substack{\forall p,q \\ (p \neq q)}\in\{1,2,\cdots,N_R\}} D_{KL}(\theta_{\#npe}(\widetilde{x}_{ip}), \theta_{\#npe}(\widetilde{x}_{iq})),$$

where $|\widetilde{x_{il}}| = N;\ \forall l \in \{1, 2, \cdots, N_R\}$. We calculate the average of the values of $\bar{D_S}_i(N,_{\#npe}, N_R)$ over the M different 16S rRNA samples:

$$\bar{D_S}(N,_{\#npe}, N_R) = \frac{1}{M}\sum_{i=1}^{M} \bar{D_S}_i(N,_{\#npe}, N_R).$$

We measure the **unrepresentativeness ($\bar{D_R}$)** of the resamples by calculating the average Kullback Leibler divergence between normalized NPE distributions for N_R resamples (N_R=10) with size N and using all the sequences in X_i for the i^{th} 16S rRNA sample:

$$\bar{D_R}_i(N,_{\#npe}, N_R) = \frac{1}{N_R} \sum_{\forall p\in\{1,2,\cdots,N_R\}} D_{KL}(\theta_{\#npe}(\widetilde{x}_{ip}), \theta_{\#npe}(X_i)),$$

where $|\widetilde{x_{il}}| = N;\ \forall l \in \{1, 2, \cdots, N_R\}$. We calculate the average over $\bar{D_R}_i(N, k)$'s for the M 16S rRNA samples:

$$\bar{D_R}(N,_{\#npe}, N_R) = \frac{1}{M}\sum_{i=1}^{M} \bar{D_R}_i(N,_{\#npe}, N_R).$$

For the experiments on the datasets presented in §3.4, we measure self-inconsistency $\bar{D_S}$ and unrepresentativeness $\bar{D_R}$ for $N_R = 10$ and $M = 10$ for $\#npe \in \{10000, 20000, 50000\}$ with sampling sizes ranging from 20 to 20000.

As shown in Figure 3.7, the obtained NPE representation in the first component will be then used for two main use cases, i.e. phenotype prediction and biomarker detection.

Phenotype prediction

We used Random Forest (RF) classifiers (Breiman 2001), which have shown a superior performance over deep neural network (deep multi-layer perceptron) and support vector machine (SVM) classifiers in phenotype classification for the size of datasets we use here (Asgari, Garakani, et al. 2018; Statnikov et al. 2013). However, the provided implementation provides deep learning and SVM classifiers as well. For the disease phenotype prediction, Random

Forest classifiers were tuned for (i) the number of decision trees in the ensemble, (ii) the number of features for computing the best node split, and (iii) the function to measure the quality of a split. We evaluate and tune the model parameter using stratified 10 fold cross-validation and optimize the classifiers for the harmonic mean of precision and recall, i.e. the F1-score, as a trade-off between precision and recall. We provide both micro- and macro-F1 metrics, which are averaged over instances and over categories, respectively.

We performed phenotype classification for a synthetic dataset (binary classification of 100 case samples and 100 control samples), a Crohn's disease dataset (binary classification of 731 Crohn's disease samples from 628 control or other diseases), and a Rheumatoid Arthritis (RA) dataset (44 RA disease subjects versus 70 control/treated/Psoriatic arthritis subjects). In order to evaluate the performance of NPE representation, we compare the classification performance of RFs over NPE features versus using OTUs, as well as k-mer features, which are considered as state-of-the-art approaches for disease phenotype prediction (Asgari, Garakani, et al. 2018; Statnikov et al. 2013).

Biomarker detection and taxonomic analysis

The designed steps in DiTaxa for detection of differently expressed markers in the phenotype of interest are shown in the light purple background in Figure 3.7:

1. The first step is finding discriminative markers between two phenotype states using false discovery rate corrected two-sided χ^2 test over the median-adjusted presence of markers in the samples. Thus if a marker is presented within a sample at least as frequent as the median frequency across samples, we consider it as present, otherwise as absent. We discard insignificant markers using a threshold for the p-value of < 0.05. For the multi-phenotype case, a one-versus-all policy is used. In addition, markers shorter than a certain threshold ($< 50bps$) will be discarded to ensure the markers are specific enough for a downstream taxonomic assignment.

2. The filtered markers go through a local BLAST (Camacho et al. 2009) with EzBioCloud database as a local reference dataset (Yoon et al. 2017), covering 62,362 quality controlled reference sequences. We assign the taxon corresponding to the Lowest Common Ancestor (LCA) of the taxa annotated for the best hits of a marker in a reference taxonomy. The markers that cannot be aligned to the references will be marked as 'Novel' markers.

3. In the third step, we remove redundant markers based on their co-occurrence information using symmetric Kullback–Leibler divergence (Kullback and Leibler 1951):

$$D_{\mathrm{KL_{sym}}}(P_m \| P_n) = \sum_i P_m(i) \, \log \frac{P_m(i)}{P_n(i)} + P_n(i) \, \log \frac{P_n(i)}{P_m(i)},$$

where P_m and P_n are respectively normalized frequency distributions of m^{th} and n^{th} markers across all samples. Using ($D_{\mathrm{KL}_{\mathrm{sym}}} = 0$) to find identical markers, split the set of markers into equivalence classes. Subsequently, from each class we pick only one representative marker with the most specific taxonomy level, which its taxonomy information is confirmed by the majority of markers within the class. The selected markers at this step are our final set of biomarkers.

4. Our approach has three main outputs, first, a taxonomic tree for significant discriminative biomarkers, where identified taxa to the positive and negative class are colored according to their phenotype (red for positive class and blue for negative class). The DiTaxa implementation for taxonomic tree generation uses a Phylophlan-based backend (Segata, Boernigen, et al. 2013). Second, a heatmap of top biomarkers occurrences in samples, where the rows denote markers and the columns are samples is generated. Such a heatmap allows biologists to obtain a detailed overview of markers' occurrences across samples (e.g., Figure 3.16 and Figure 3.17). The heatmap shows number of distinctive sequences hit by each biomarker in different samples and stars in the heatmap denote hitting unique sequences, which cannot be analyzed by OTU clustering approaches. The third output is a list of novel markers for further analysis of the potential novel organisms.

To compare the performance of our approach with a standard workflow (defined in §3.4) for real datasets, we used the scientific literature as the ground-truth. We extracted a list of organisms which are experimentally identified by previous studies to be associated with or cause periodontal disease. Then we evaluate the recall for DiTaxa and the standard workflow in the detection of the confirmed organisms.

To quantify the performance of DiTaxa in a known synthetic setting versus the standard pipeline, we generated two high-dimensional synthetic datasets denoted as 'case' and 'control', as described in §3.4. The description of the standard pipeline is provided in §3.4 generating a list of significant differently expressed OTUs for both phenotypes. We then compare the significantly enriched OTU sequences (FDR corrected P values < 0.05, LEfSe) and significant (FDR < 0.05) subsequences determined by DiTaxa as biomarkers with the ground-truth 16S V4 region using global nucleotide alignment with blastn v. 2.7.1+ with parameters "$-perc\_identity 100 - ungapped$". We quantified the number of false positives (FP), true positives (TP), false negatives (FN) and true negatives (TN) based on the presence of significant alignments of potential biomarkers sequences (OTU or DiTaxa markers) to each of the 500 differentially expressed GG 16S regions. TPs were calculated as the number of GG sequences ($n = 500$) with at least one marker hit from the positive marker list. FNs were calculated using case GG sequences ($n = 500$) without at least one marker hit from the marker collection found to be significantly enriched in the case set. TNs are the number of control GG sequences ($n = 500$) with at least one marker hit from the marker list that are significant in the control set. FPs were quantified as the number of low abundant GG sequences ($n = 500$) without at least one marker hit from the set of markers found in the

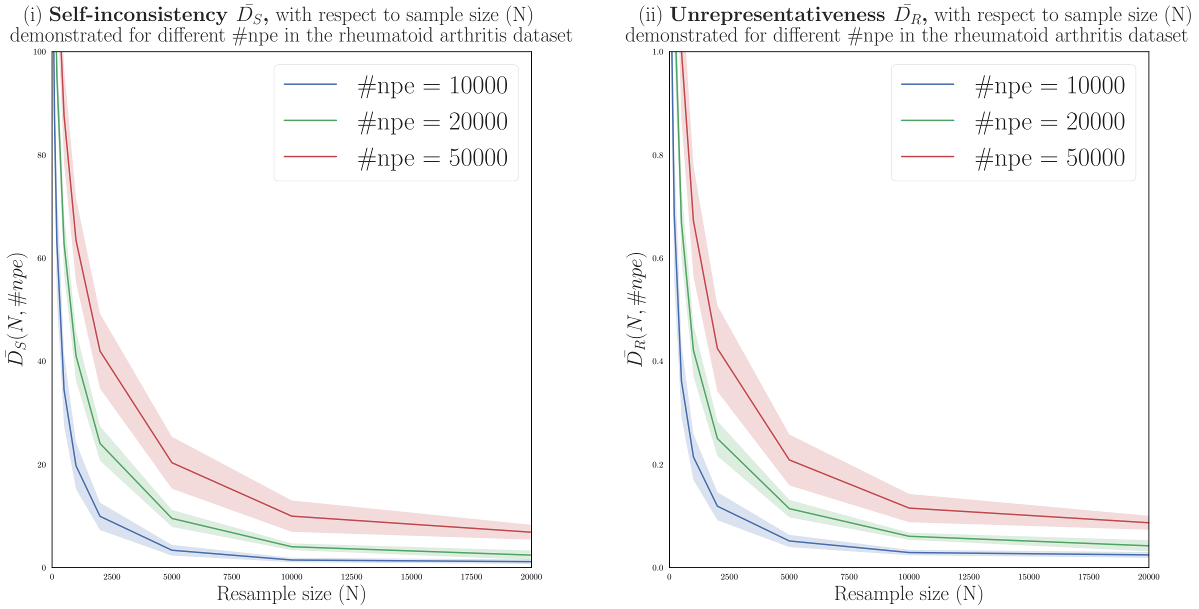


Figure 3.8. Measuring (i) self-inconsistency ($\bar{D_S}$), and (ii) unrepresentativeness ($\bar{D_R}$) for the arthritis disease dataset. Each point represents an average of 100 resamples belonging to 10 randomly selected 16S rRNA samples. Higher vocabulary size require higher sampling rates to produce self-consistent and representative samples.

control sample. Recall was calculated as $TP/(TP + FN)$ while precision was calculated as $TP/(TP + FP)$.

Results

Phenotype prediction

Bootstrapping for sample size selection

We picked a stable sample size for each NPE vocabulary size based on the output of bootstrapping in phenotype prediction. Each point in Figure 3.8 represents the average of 100 ($M \times N_R$) resamples belonging to M randomly selected 16S rRNA samples, each of which is resampled $N_R = 10$ times. As shown in Figure 3.8, a larger vocabulary size require higher sampling rates to produce self-consistent and representative representations. As the structure of the curve does not vary a lot from dataset to dataset, to avoid redundancy, we only show the bootstrapping results for the rheumatoid arthritis dataset.

The classification results for different NPE vocabulary sizes on the synthetic, Crohn's disease, and rheumatoid arthritis datasets using RF classifiers are presented in Table 3.8. All methods could reliably predict the affected cases in the synthetic dataset without any error. For the Crohn's disease dataset k-mers with the MicroPheno approach (Asgari, Garakani, et al. 2018) achieved a slightly better prediction performance while using NPE and OTU

Table 3.8. Results for NPEs, OTUs, and k-mer features in performing phenotype classification over the synthetic, rheumatoid arthritis, and Crohn's disease datasets in a 10-fold cross-validation framework using random forest classifiers.

Dataset	Representation	Classifier	Feature-size	Resample size	Micro-metrics (averaged over samples)			Macro-metrics (averaged over classes)		
					Precision	Recall	F1	Precision	Recall	F1
Synthetic	NPE	RF	10000	5000	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
			20000	10000	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
			50000	All sequences	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	6-mer		4096	5000	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	OTU		1041	-	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Crohn's disease	NPE	RF	10000	5000	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.73 ± 0.04	0.73 ± 0.04
			20000	10000	0.75 ± 0.04	0.75 ± 0.04	0.75 ± 0.04	0.75 ± 0.04	0.74 ± 0.04	0.74 ± 0.04
			50000	All sequences	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04
	6-mer		4096	5000	0.76 ± 0.04	0.76 ± 0.04	0.76 ± 0.04	0.76 ± 0.04	0.75 ± 0.04	0.75 ± 0.04
	OTU		9511	-	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04	0.74 ± 0.04
Rheumatoid Arthritis	NPE	RF	10000	5000	0.78 ± 0.1	0.78 ± 0.1	0.78 ± 0.1	0.78 ± 0.11	0.76 ± 0.13	0.76 ± 0.12
			20000	10000	0.78 ± 0.09	0.78 ± 0.09	0.78 ± 0.09	0.78 ± 0.11	0.76 ± 0.11	0.75 ± 0.11
			50000	All sequences	0.78 ± 0.1	0.78 ± 0.1	0.78 ± 0.1	0.79 ± 0.11	0.76 ± 0.13	0.75 ± 0.12
	6-mer		4096	5000	0.78 ± 0.1	0.78 ± 0.1	0.78 ± 0.1	0.78 ± 0.12	0.76 ± 0.12	0.76 ± 0.12
	OTU		949	-	0.69 ± 0.12	0.69 ± 0.12	0.69 ± 0.12	0.73 ± 0.16	0.65 ± 0.12	0.65 ± 0.11

features achieved the same macro-F1 of 0.74. In rheumatoid arthritis prediction, NPE and k-mers achieved a macro-F1 of 0.76, outperforming the use of OTU features by 11 percent. Changes in sample size did not substantially affect the prediction performance, suggesting sufficiency of shallow sub-samples in phenotype prediction using the NPE representation (Table 3.8).

Biomarker detection and taxonomic analysis

Marker detection results for synthetic data

In the biomarker detection for the synthetic dataset, DiTaxa did not report erroneous (neither FN nor FP) biomarkers, which resulted in a recall and precision value of 1. In comparison, the biomarkers detected with a standard pipeline, using OTU clustering and LEfSe, included 51 false negative and 47 false positive instances, resulting in a recall and precision value of 0.898 and 0.905, respectively (Table 3.9). This evaluation demonstrated the superiority of DiTaxa for biomarker discovery compared to OTU-based approaches in both recall and precision.

Table 3.9. The results of DiTaxa and the standard pipeline (STDP) in marker detection for the synthetic dataset.

Method	Precision	Recall	F1
DiTaxa	1	1	1
STDP	0.905	0.898	0.901

Table 3.10. The results of DiTaxa and the standard pipeline (STDP) in marker detection in comparison with literature of periodontal disease.

Method	True Positive Count	Recall
DiTaxa	13 out of 29	0.59
STDP	3 out of 29	0.10

Biomarker detection results for periodontal disease

We next assessed the performance of DiTaxa and a standard pipeline (STDP; section 3.4) in detecting otherwise confirmed taxa for periodontal disease (Table 3.11). DiTaxa performed better in the detection of relevant taxa (Table 3.10). Of 29 taxa identified as relevant in other studies, 17 were detected by DiTaxa, while the standard approach detected only 3 from the same dataset. Notably, experimentally verified taxa shown to alter the disease phenotype in mouse models, *Fusobacterium nucleatum* (Polak et al. 2009) and *Porphyromonas gingivalis* (Kimura et al. 2000; Polak et al. 2009; Hajishengallis et al. 2011), were only detected by DiTaxa. Since periodontitis is a polymicrobial disease and the oral biofilms are extremely diverse (Perez-Chaparro et al. 2014), detecting all relevant taxa confirmed by the literature from a single dataset is not feasible. For instance, A. actinomycetemcomitans is specifically associated with juvenile aggressive periodontists in Moroccan population, which can be hardly found in the population from Turkey (Jorth et al. 2014). However, relative comparison of recall for different methods on the same dataset is still meaningful. A higher recall of DiTaxa in confirming the literature links, in comparison with a standard pipeline shows that DiTqxa can be more accurate in detecting disease-specific biomarkers. A detailed comparison of the predicted taxa with different methods and taxa with confirmed links by the literature is also shown in Figure 3.9. The red color shows the disease associated taxa found by DiTaxa and the blue color indicates the up-regulated taxonomy in the healthy samples. The up-regulated organisms found by standard pipeline are colored in yellow and down-regulated organisms are colored to green. The intersection of DiTaxa and the standard approach is colored in orange for up-regulation and cyan denotes for the consensus of methods in down-regulation.

Taxonomy of discriminative biomarkers for rheumatoid arthritis

Comparative taxonomic visualization of detected differentially expressed markers for DiTaxa and a common workflow are shown in Figure 3.10 for samples from patients with untreated rheumatoid arthritis (new onset RA) versus healthy individuals. Taxa predicted by DiTaxa for samples from patients with untreated rheumatoid arthritis (new onset RA) versus healthy individuals had *Prevotella copri* as the most significantly ranked, which was confirmed based on shotgun metagenome analysis and in mouse experiments (Scher, Sczesnak, et al. 2013), while the standard workflow only predicted the genus of this taxon as relevant (Scher, Sczesnak, et al. 2013). DiTaxa also predicted *Prevotella stercorea* as implicated in new onset RA.

The DiTaxa results for patient samples from several other diseases versus healthy individuals are provided in Figure 3.11 (for CD versus healthy), Figure 3.13 (for indeterminate colitis versus healthy), Figure 3.12 (for ulcerative colitis versus healthy), Figure 3.15 (for treated rheumatoid arthritis versus healthy), Figure 3.14 (for psoriatic versus healthy).

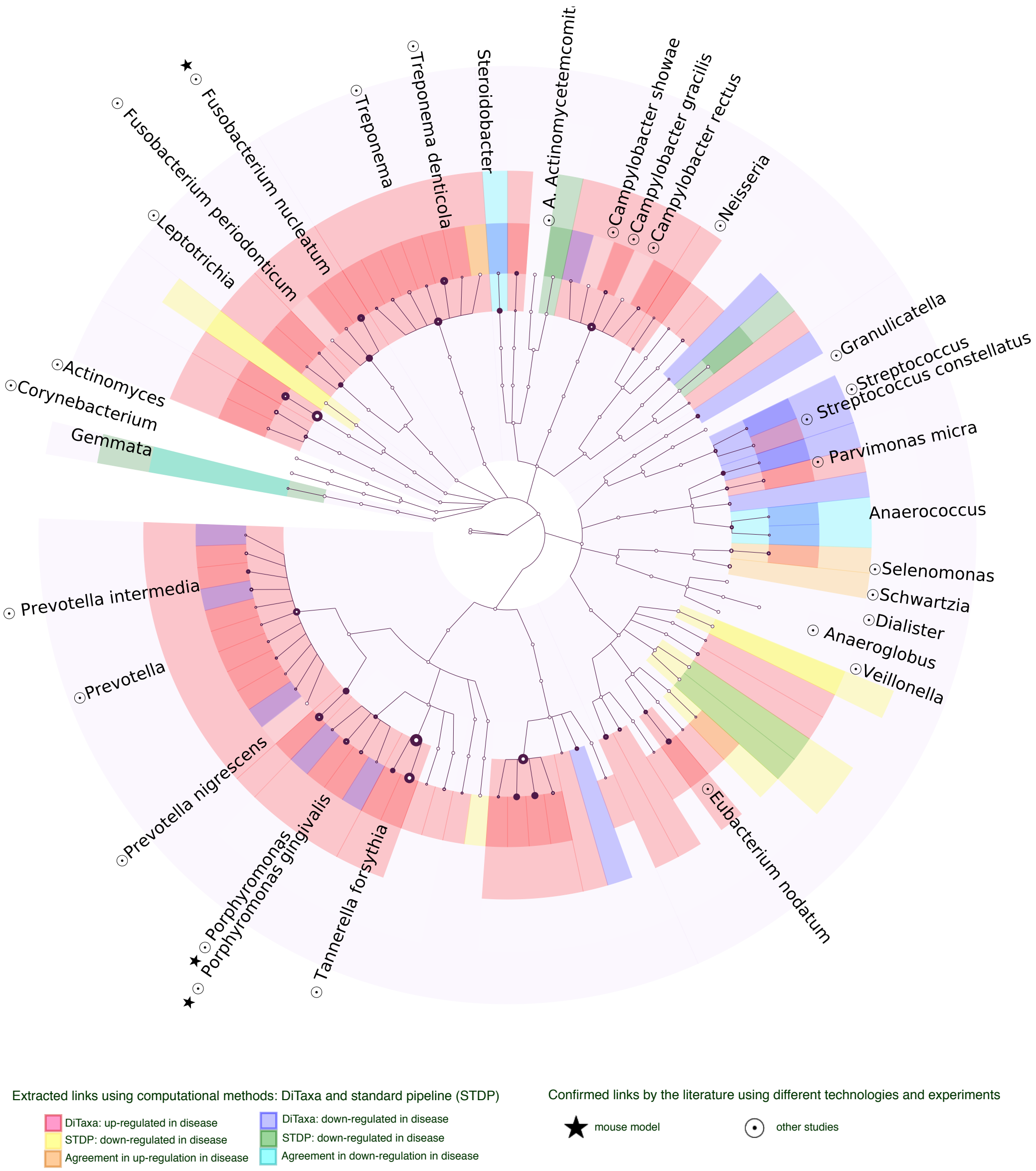

★ ⊙ Fusobacterium nucleatum
⊙ Fusobacterium periodonticum
⊙Treponema
⊙Treponema denticola
Steroidobacter
⊙ A. Actinomycetemcomit.
⊙Campylobacter showae
⊙Campylobacter gracilis
⊙Campylobacter rectus
⊙Neisseria
⊙Leptotrichia
⊙Granulicatella
⊙Streptococcus
⊙ Streptococcus constellatus
⊙ Parvimonas micra
⊙Actinomyces
⊙Corynebacterium
Gemmata
Anaerococcus
⊙Selenomonas
⊙Schwartzia
⊙Dialister
⊙ Anaeroglobus
⊙Veillonella
⊙ Prevotella intermedia
⊙Prevotella
⊙Prevotella nigrescens
⊙Eubacterium nodatum
★ ⊙ Porphyromonas
★ ⊙ Porphyromonas gingivalis
⊙ Tannerella forsythia
Extracted links using computational methods: DiTaxa and standard pipeline (STDP)
🟥 DiTaxa: up-regulated in disease
🟨 STDP: down-regulated in disease
🟧 Agreement in up-regulation in disease
🟦 DiTaxa: down-regulated in disease
🟩 STDP: down-regulated in disease
🟦 Agreement in down-regulation in disease
Confirmed links by the literature using different technologies and experiments
★ mouse model
⊙ other studies

Figure 3.9. Taxonomy of differently expressed markers for samples from patients with periodontal disease versus healthy patients. This tree diagram illustrates the hierarchical relationships among the detected organisms. On top of the tree structure, the colors visualize different metadata exist for each organism. The red color shows the disease associated taxa found by DiTaxa and the blue color indicates the up-regulated taxonomy in the healthy samples. The up-regulated organisms found by standard pipeline (STDP) are colored in yellow and down-regulated organisms are colored to green. The intersection of DiTaxa and the standard approach is colored in orange for up-regulation and cyan denotes for the consensus of methods in down-regulation. When two methods disagree the organism is colored in gray. The node size is proportional to the number of markers confirming the taxa and the border boldness is proportional to the negative log of the markers' p-value. The taxa names are only provided for the nodes that are either confirmed by the literature or the nodes denoting agreements of DiTaxa and STDP. Zooming in the electronic version is possible for a better see of the details.

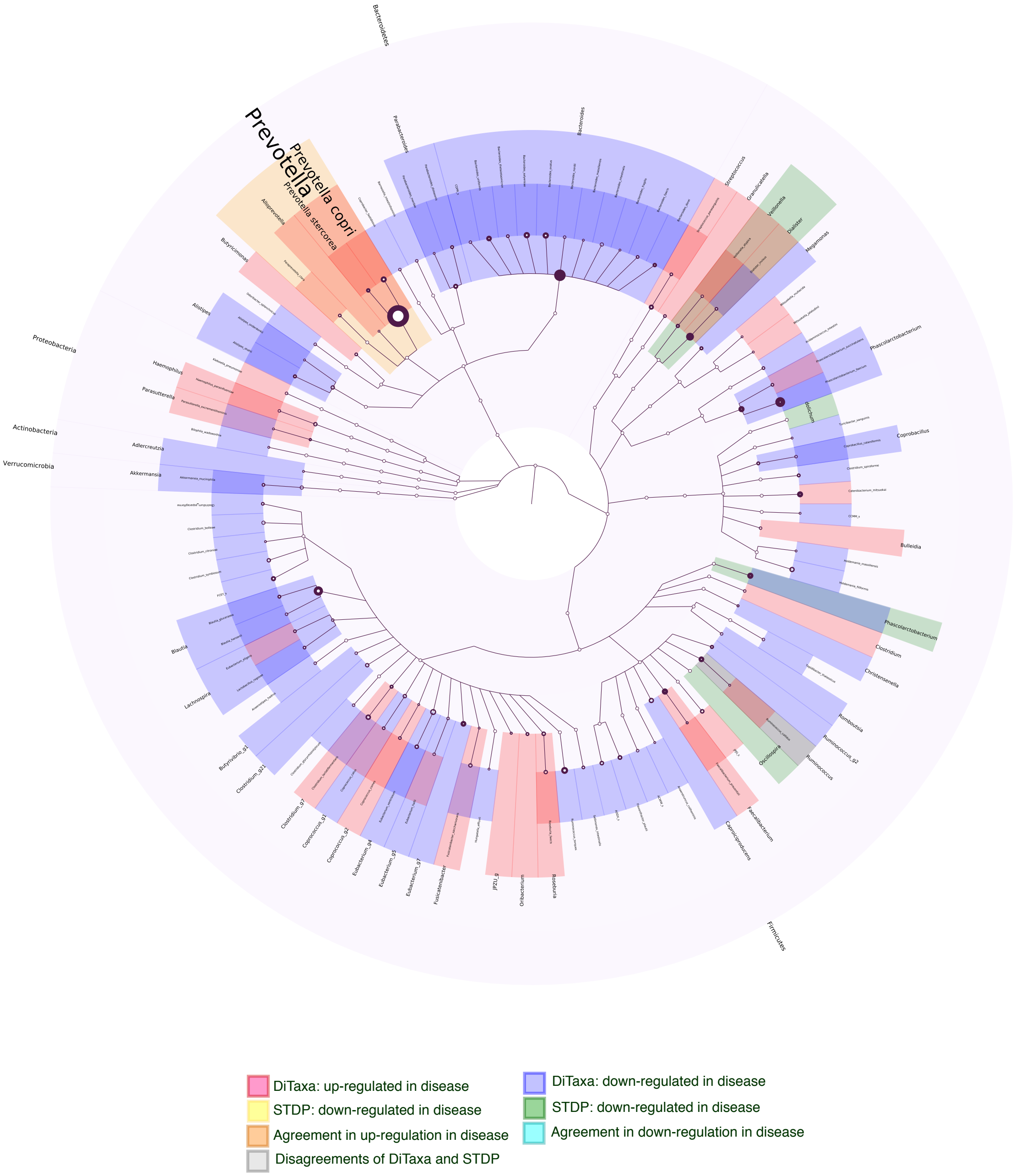


Figure 3.10. Taxonomy of differently expressed markers for new onset rheumatoid arthritis versus healthy samples. The red color shows the disease-associated taxa found by DiTaxa and the blue color denotes the up-regulated taxa in the samples from healthy individuals. The up-regulated organisms found by the standard pipeline are colored in yellow and down-regulated organisms are colored green. The agreements of DiTaxa and the standard approach are colored in orange for up-regulation and cyan denotes the consensus of both methods in down-regulated taxa. When the two methods disagree, the taxon is colored in gray. The node size is proportional to the number of markers confirming the taxa and the border boldness is proportional to the negative log of the markers' p-value. Zooming in the electronic version is possible for a better see of the details.

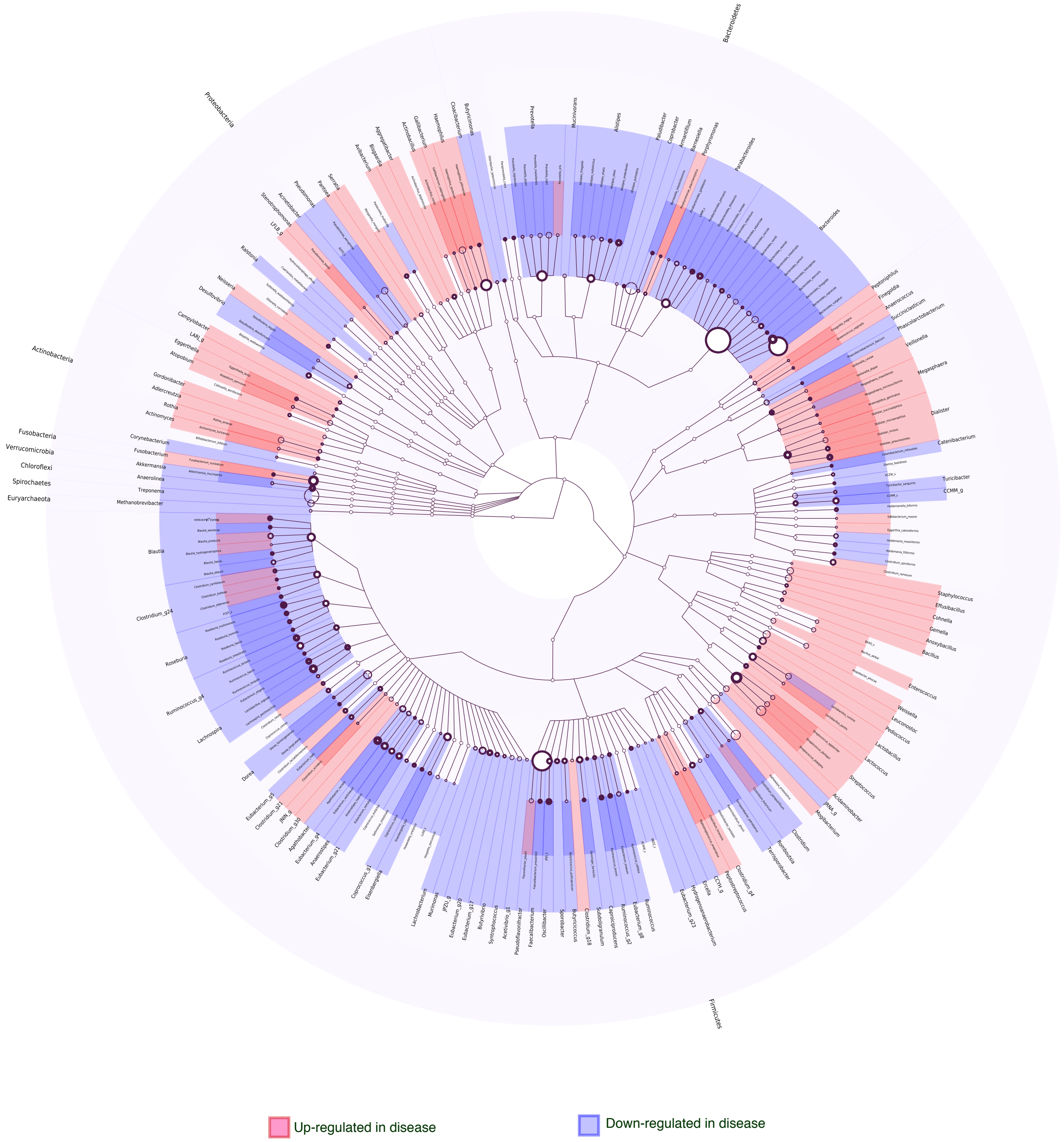


Figure 3.11. Taxonomy of differently expressed markers for Crohn's disease versus healthy using DiTaxa. The red color shows the disease associated taxa found by DiTaxa and the blue color shows the up-regulated taxonomy in the healthy samples. Zooming in the electronic version is possible for a better see of the details.

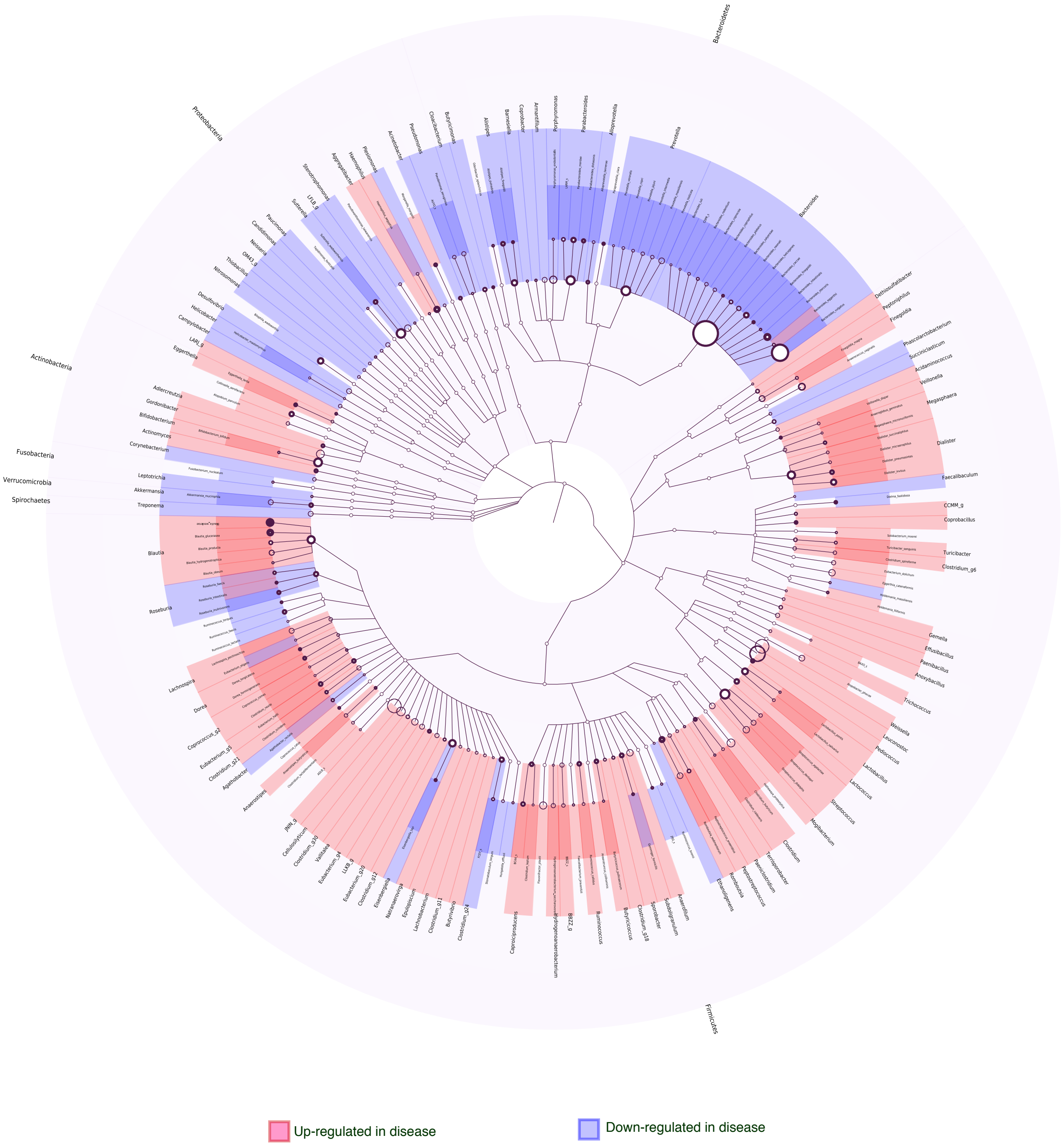


Figure 3.12. Taxonomy of differently expressed markers for ulcerative colitis disease versus healthy using DiTaxa. The red color shows the disease associated taxa found by DiTaxa and the blue color shows the up-regulated taxonomy in the healthy samples. Zooming in the electronic version is possible for a better see of the details.

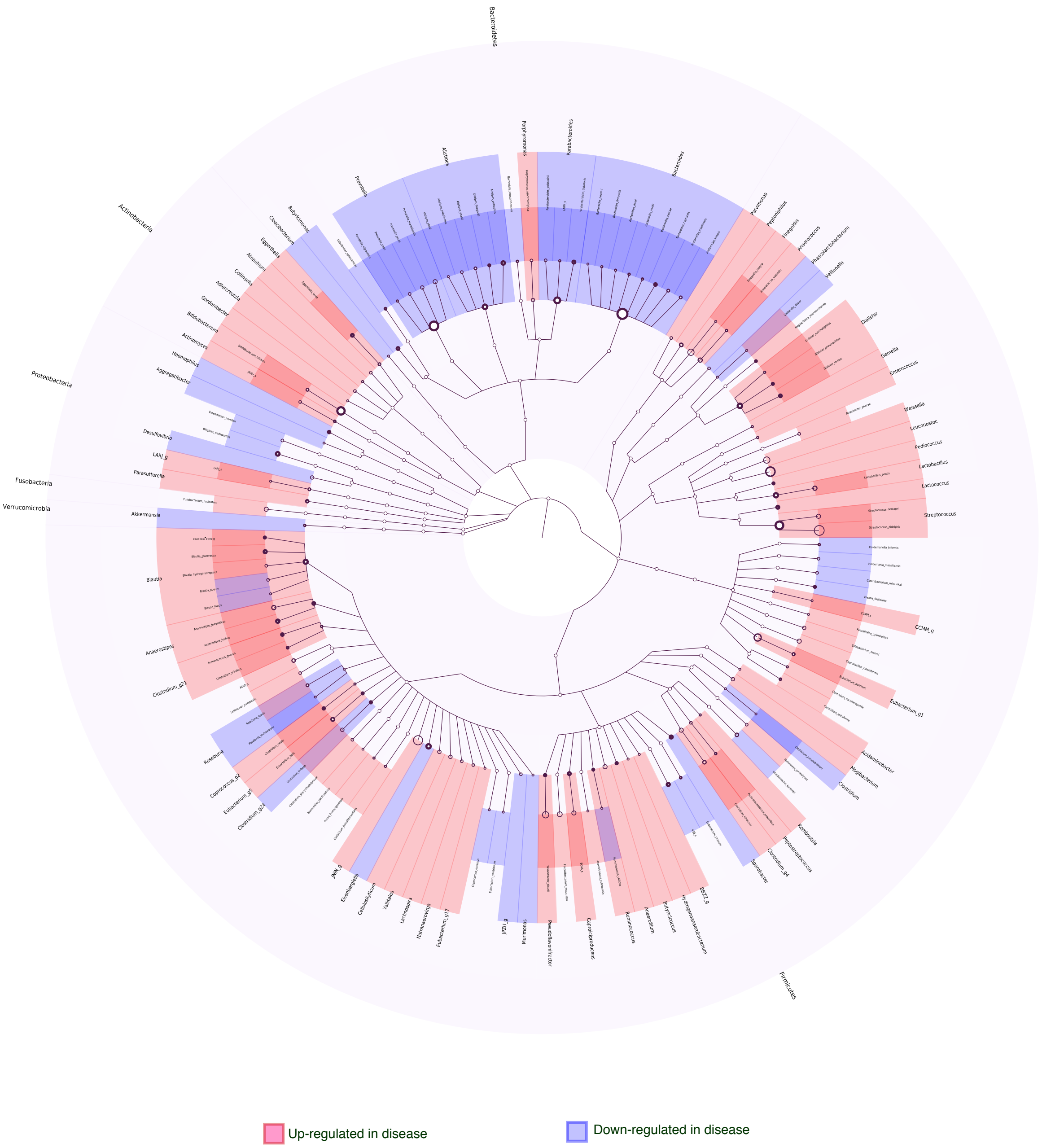


Figure 3.13. Taxonomy of differently expressed markers for indeterminate colitis disease versus healthy using DiTaxa. The red color shows the disease associated taxa found by DiTaxa and the blue color shows the up-regulated taxonomy in the healthy samples. Zooming in the electronic version is possible for a better see of the details.

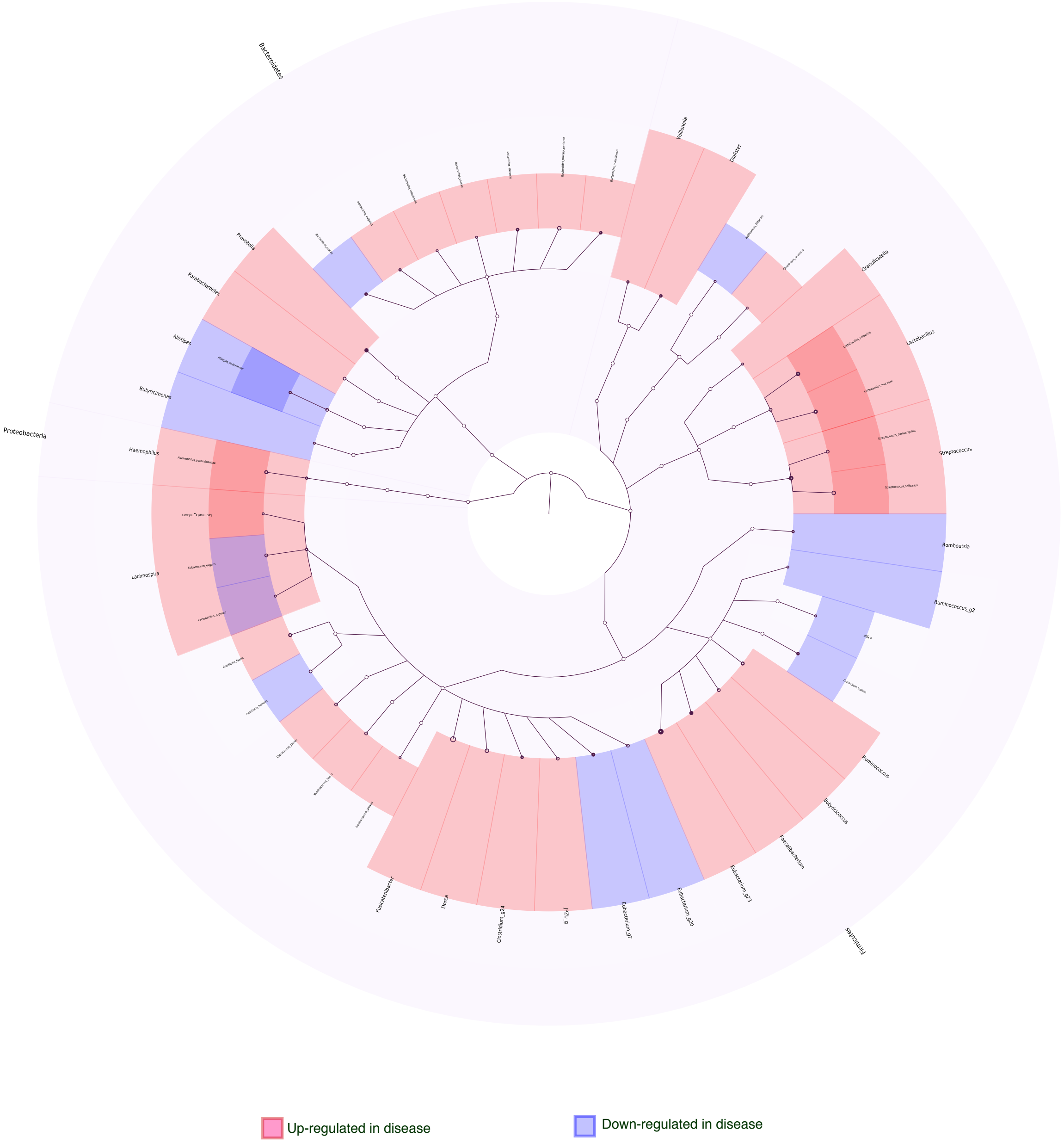


Figure 3.14. Taxonomy of differently expressed markers for psoriatric disease versus healthy using DiTaxa. The red color shows the disease associated taxa found by DiTaxa and the blue color shows the up-regulated taxonomy in the healthy samples. Zooming in the electronic version is possible for a better see of the details.

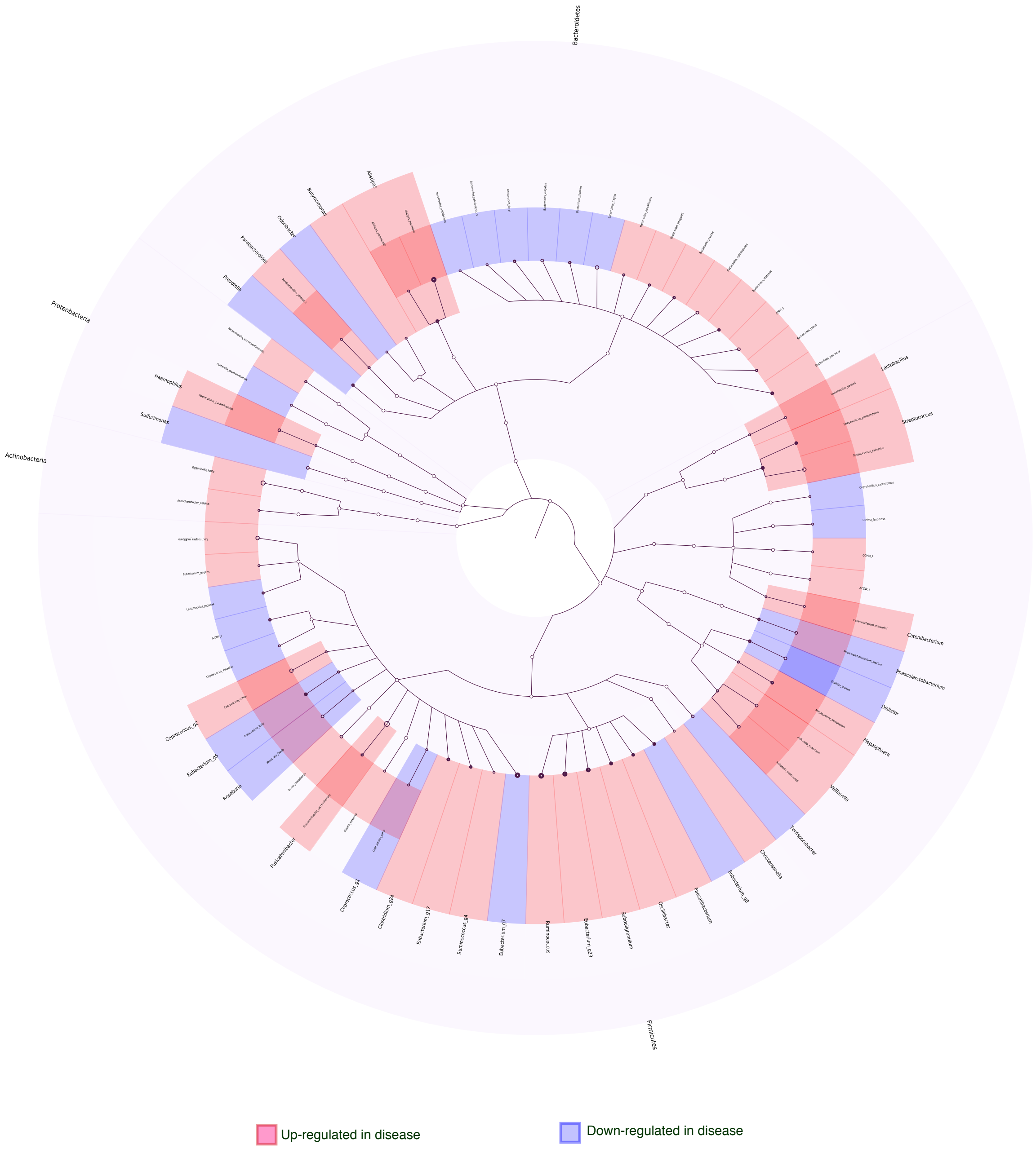


Figure 3.15. Taxonomy of differently expressed markers for treated rheumatoid Arthritis versus healthy samples using DiTaxa. The red color shows the treated associated taxa found by DiTaxa and the blue shows the up-regulated taxonomy in the healthy samples. Zooming in the electronic version is possible for a better see of the details.

Table 3.11. Comparison of a standard pipeline (STDP) and DiTaxa performance in detecting 29 taxa with confirmed links to periodontitis. The upwards arrow denotes upregulation in the diesease group compared to samples from healthy subjects. A checkmark denotes presence of marker by DiTaxa method.

Literature information				Detection results	
Genus	Species	Direction	Description and reference	DiTaxa	STDP
Actinomyces	n.a.	↓	16S rRNA sequencing of supragingival plaque samples, t-test (Bo Liu et al. 2012), Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012)	not found	not found
Aggregatibacter	A. actinomycetemcomitans	↑	Used for mixed infections for murine back abscess model (Oz and Puleo 2011)	not found	not found
Anaeroglobus	n.a.	↑	Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012)	not found	not found
Corynebacterium	n.a.	↓	Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012)	not found	not found
Campylobacter	C. gracilis	↑	16S rRNA sequencing (Teles et al. 2013)	✓	not found
Campylobacter	C. rectus	↑	16S rRNA sequencing (Teles et al. 2013)	not found	not found
Campylobacter	C. showae	↑	16S rRNA sequencing (Teles et al. 2013)	not found	not found
Dialister	n.a.	↑	Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012)	not found	not found
Eubacterium	E. nodatum	↑	16S rRNA sequencing (Teles et al. 2013)	✓	not found
Fusobacterium	F. nucleatum	↑	Mouse model (Polak et al. 2009), Whole genomic DNA probes of subgingival plaque samples (Socransky et al. 1998), and dot blot hybridization (Schlafer et al. 2010)	✓	not found
Fusobacterium	F. periodonticum	↑	microbial co-localization in images of subgingival biofilm species (Schillinger et al. 2012)	not found	not found
Gamella	n.a.	↓	16S study of plague samples from healthy individuals (Aas et al. 2005)	not found	not found
Granulicatella	n.a.	↓	16S study of plague samples from healthy individuals (Aas et al. 2005), Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012)	not found	not found
Leptotrichia	n.a.	↑	16S rRNA sequencing of supragingival plaque samples, t-test (Bo Liu et al. 2012)	✓	not found
Neisseria	n.a.	↓,↑	Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012) (↓), Whole genomic DNA probes of subgingival plaque samples(Ximenez-Fyvie et al. 2000)(↑)	✓(↑)	not found
Parvimonas	P. micra	↑	16S rRNA sequencing (Perez-Chaparro et al. 2014)	✓	not found
Porphyromonas	n.a.	↑	Mouse model (Hajishengallis et al. 2011) and Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012)	✓	not found
Porphyromonas	P. gingivalis	↑	Mouse model (Kimura et al. 2000; Polak et al. 2009), Whole genomic DNA probes of subgingival plaque samples(Socransky et al. 1998), Used for monomicrobial infections for murine back abscess model (Oz and Puleo 2011)	✓	not found
Prevotella	n.a.	↑	16S rRNA sequencing of supragingival plaque samples, t-test (Bo Liu et al. 2012), Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012)	✓	not found
Prevotella	P. intermedia	↑	Whole genomic DNA probes of subgingival plaque samples (Socransky et al. 1998) and dot blot hybridization (Schlafer et al. 2010)	✓	not found
Prevotella	P. nigrescens	↑	16S rRNA sequencing (Teles et al. 2013)	not found	not found
Schwartzia	n.a.	↑	Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012)	✓	✓
Selenomonas	n.a.	↑	Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012)	✓	✓
Streptococcus	n.a.	↓	16S rRNA sequencing of supragingival plaque samples, t-test (Bo Liu et al. 2012), 16S study of plague samples from healthy individuals (Aas et al. 2005)	✓	not found
Streptococcus	S. constellatus	↑	16S rRNA sequencing of supragingival plaque samples, t-test (Abusleme et al. 2013)	✓	not found
Tannerella	T. forsythia	↑	Whole genomic DNA probes of subgingival plaque samples (Socransky et al. 1998), 16S rRNA squencing (Ledder et al. 2007)	✓	not found
Treponema	n.a.	↑	Periodontal disease vs. healthy control gingiva, t-tests (Scher, Ubeda, et al. 2012)	✓	✓
Treponema	T. denticola	↑	Whole genomic DNA probes of subgingival plaque samples (Socransky et al. 1998), nested PCR(Perez-Chaparro et al. 2014)	✓	not found
Veillonella	n.a.	↓	16S study of plague samples from healthy individuals (Aas et al. 2005)	not found	not found

Biomarker heatmaps

Visualization of the occurrence pattern of the identified biomarkers is another output of DiTaxa. Examples of such a visualization for rheumatoid arthritis (Figure 3.16) and periodontal disease (Figure 3.17) are provided. The rows represent inferred biomarker sequences and are sorted based on the taxonomic marker assignments. The columns represent patient samples and are sorted firstly based on their phenotype and secondly based on their pattern similarity. 'Novel' organisms are shown in the top rows, denoting the markers that could not be aligned to any reference sequence and are therefore potentially novel taxa. The cell colors on the heatmap show the percentage of distinct marker sequences matching a biomarker per sample on a log scale. Markers targeting a single 16S sequence only are marked by a star.

The plots clearly show the varying "generality" of the inferred marker sequences, with some matching only to unique 1S sequences, and others found across larger numbers, indicating representation of different levels of evolutionary relatedness of the underlying targeted organisms. For instance, of the inferred markers assigned *Prevotella copri*, while some markers match multiple distinct 16S genes across patient samples, indicating the presence of strain-level diversity evident from 16S within this species, while other markers targeting predominantly single 16S copies across patients samples within this species, indicating the existence of disease-associated subspecies diversity, that can be discovered with this technique.

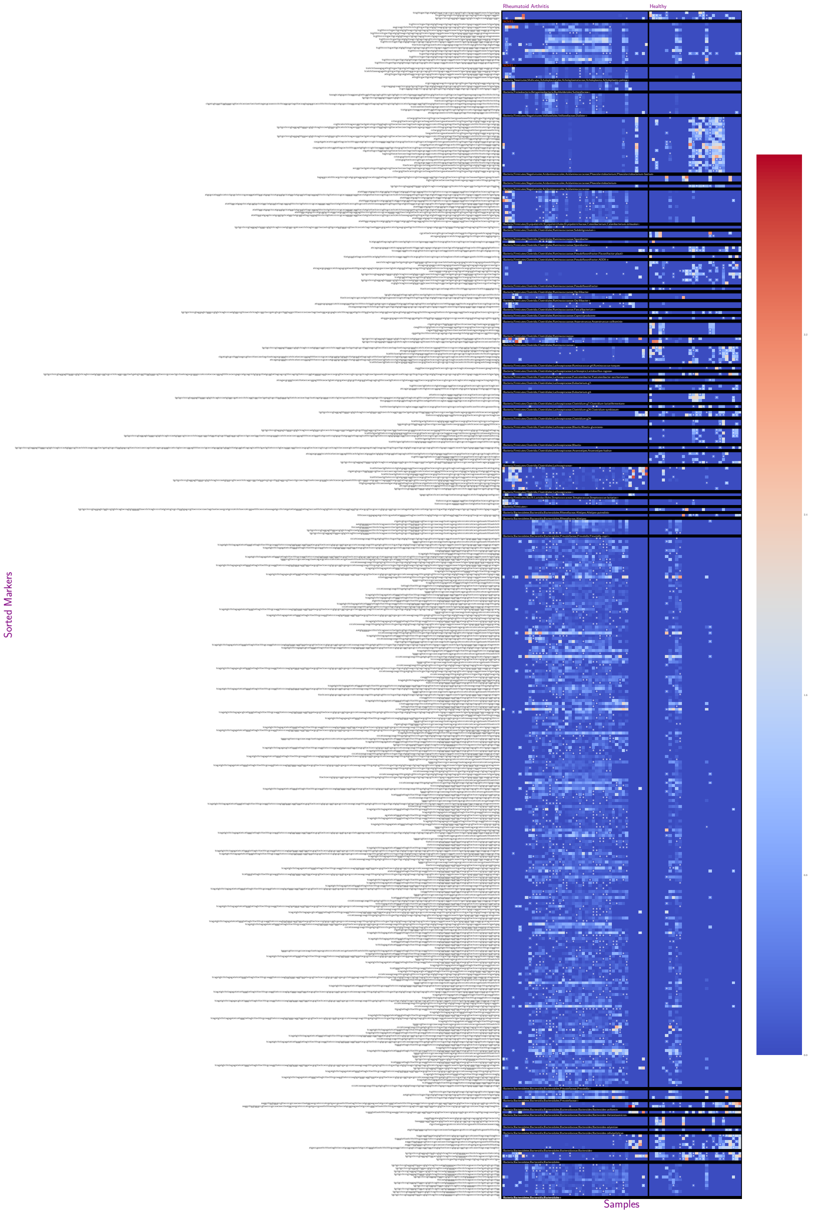


Figure 3.16. Heatmap of markers occurrences across samples for rheumatoid arthritis. The rows represent inferred biomarker sequences and are sorted based on the taxonomic marker assignments. The columns represent patient samples and are sorted firstly based on their phenotype and secondly based on their pattern similarity. 'Novel' organisms are shown in the top rows, denoting the markers that could not be aligned to any reference sequence and are therefore potentially novel taxa. The cell colors on the heatmap show the percentage of distinct marker sequences matching a biomarker per sample on a log scale. Markers targeting a single 16S sequence only are marked by a star. Zooming in the electronic version is possible for a better see of the details.

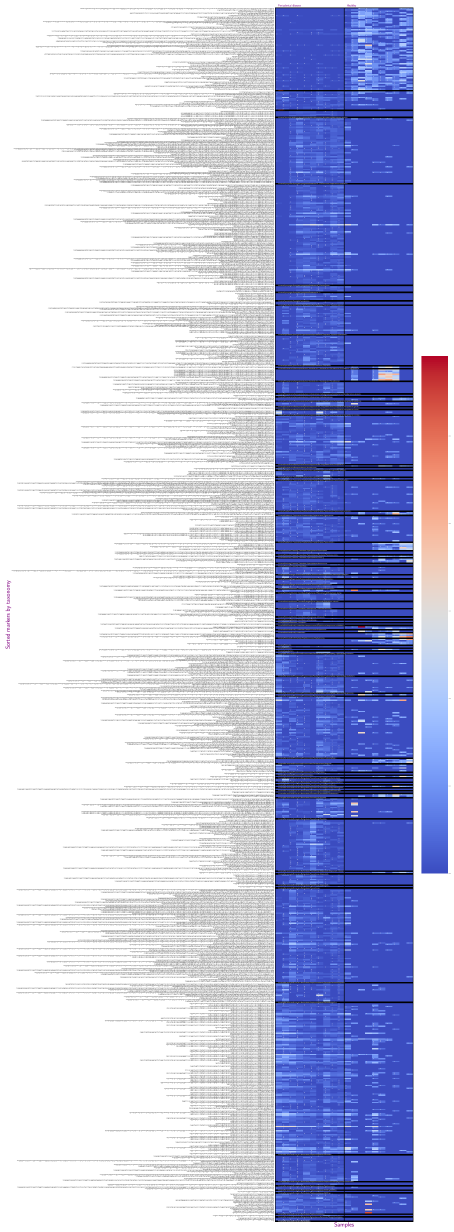


Figure 3.17. Heatmap of markers occurrences across samples for periodontal disease. The rows represent inferred biomarker sequences and are sorted based on the taxonomic marker assignments. The columns represent patient samples and are sorted firstly based on their phenotype and secondly based on their pattern similarity. 'Novel' organisms are shown in the top rows, denoting the markers that could not be aligned to any reference sequence and are therefore potentially novel taxa. The cell colors on the heatmap show the percentage of distinct marker sequences matching a biomarker per sample on a log scale. Markers targeting a single 16S sequence only are marked by a star. Zooming in the electronic version is possible for a better see of the details.

Table 3.12. Runtimes of DiTaxa and a common 16S processing pipeline for the periodontal, synthetic, rheumatoid arthritis and IBD datasets. The (‖) sign denotes steps that are parallelized and run on multiple cores (here 20 cores). STDP stands for standard pipeline, using OTU clustering and LEfSe for marker detection.

Dataset	Number of samples	Segmentation or clustering		Representation Creation		Biomarker detection		**Total**	
		DiTaxa	STDP (‖)	DiTaxa (‖)	STDP	DiTaxa(‖)	STDP (‖)	DiTaxa	STDP
Periodontal dataset	20	3,37 min	0,34 min	1,84 min	0,43 min	1,58 min	0,10 min	**6,79 min**	**0,87 min**
Rheumatoid Arthritis dataset	114	23,86 min	1,33 min	0,61 min	5,10 min	1,73 min	0,11 min	**26,19 min**	**6,54 min**
Synthetic dataset	200	19 min	0.81 min	1,8 min	0,78 min	1,76 min	0,55 min	**22,56 min**	**2,14 min**
IBD dataset	1359	82,05 min	23,52 min	9,13 min	5,66 min	2,75 min	356,48 min	**93,93 min**	**385,66 min**

Runtime analysis

To assess computational efficiency, we compared the runtimes of DiTaxa versus the standard workflow (Table 3.12). For both DiTaxa and the standard workflow 20 cores were used in computations. Workflow parts that could be parallelized for both pipelines are denoted with "‖" in Table 3.12. The bottleneck for DiTaxa computation is the segmentation training, which cannot be parallelized. However, the segmentation needs to be trained only once for a dataset and then any combinations of phenotype analysis can use the trained segmentation and the subsequent representation. Although the standard pipeline for datasets of less than 200 samples has been few minutes faster than DiTaxa, DiTaxa can run faster for the dataset of 1359 samples (total of 93,93 min), while the standard pipeline tool 385,66 min using the same computational setting.

Discussion and conclusions

We describe DiTaxa, a method implementing a new paradigm for host disease status prediction and biomarker detection from 16S rRNA amplicon data. The main distinction of this approach from existing methods is substituting standard OTU-clustering (Robert C Edgar 2013) or sequence-level analysis (Callahan et al. 2016) by segmenting 16S rRNA reads into the most frequent variable-length subsequences of a dataset. The proposed sequence segmentation, called Nucleotide-pair Encoding, is an unsupervised approach inspired by Byte-pair Encoding, a data compression algorithm that recently became popular in deep natural language processing. The identified subsequences represent commonly occurring sequence portions, which we found to be distinctive for taxa at varying evolutionary distances and highly informative for predicting host disease phenotypes. We compared the performance of DiTaxa to the state-of-the-art in disease phenotype prediction and biomarker detection, using human 16S datasets from metagenomic samples of periodontal, rheumatoid arthritis, and inflammatory bowel diseases, as well as a synthetic benchmark dataset. DiTaxa identified 17 of 29 taxa with confirmed links to periodontitis (recall= 0.59), while the OTU-based approach could only detect 3 of 29 organisms (recall= 0.10). In addition, we show that for the rheumatoid arthritis dataset, machine-learning classifiers trained to predict host disease phenotypes based on the NPE representation substantially outperformed OTU features (macro-F1 =0.76 compared to 0.65) and performed competitively for Crohn's disease and synthetic datasets. Taxa predicted by DiTaxa for samples from patients with untreated rheumatoid arthritis (new onset RA) versus healthy individuals had *Prevotella copri* as the most significantly ranked, which was confirmed based on shotgun metagenomic analysis and in mouse experiments (Scher, Sczesnak, et al. 2013), while the standard workflow only predicted the genus of this taxon as relevant (Scher, Sczesnak, et al. 2013). Due to the alignment- and reference free nature, DiTaxa can efficiently run on large datasets. The full analysis of a large 16S rRNA dataset of 1359 samples required ≈1.5 hours, where the standard pipeline took ≈6.5 hours with the same number of cores (20 cores). Although on smaller datasets the conventional workflow was faster than DiTaxa, the run-time difference of less than 30 minutes for those settings is worth the performance gain in phenotype prediction and biomarker detection. The applications of NPE representation are not limited to 16S rRNA data and it can be also applied to shotgun metagenomics or any other biological sequences to infer intrinsic features from data, instead of using parameter-dependent representations. Taken together, DiTaxa seems to provide a better solution for biomarker and phenotype detection than OTU-based methods. It thus could contribute to a better understanding of the microbial organisms associated with microbiome-related diseases and the development of personalized diagnostics and therapy procedures.

3.5 Summary of contributions in genomics/metagenomics

We start the investigation on the use of language-agnostic representations in a genomics setting by proposing short k-mers as computationally inexpensive while being effective in the phenotype prediction in comparision with conventional gene presence/absence and SNPs features. However, the main focus of our contribution has been on metagenomics, which is a more complex problem setting. Accurately resolving taxon-host disease relationships is required to elucidate the underlying functional mechanisms of taxon-host interactions in microbiome-linked diseases and to establish diagnosis and therapy options in precision medicine. As a result, accurate detection of disease phenotype and disease-associated biomarkers are among prominent problems in microbial informatics. Although shotgun metagenomic sequencing provides a better resolution in taxonomic assignment, 16S rRNA amplicon data, due to its low cost is still the most used data type in microbiome studies to date (Pollock et al. 2018). After 16S rRNA sequencing, reads are typically clustered based on their sequence similarity to each other and the resulting clusters are referred to as operational taxonomic units (OTUs) having several disadvantages in the taxonomic assignment. An alternative solution is the analysis of individual 16S rRNA gene sequences (Callahan et al. 2016; Amir et al. 2017; Nearing et al. 2018), which is computationally challenging. In order to address accurate identifying the host phenotype/environment as well as the distinctive taxa for microbiome related diseases or phenotype. We proposed two OTU free solutions within the framework of this dissertation that are the state-of-the-art approaches to date: **(i) MicroPheno** for accurate and rapid microbial host-phenotype/environment prediction as well as **(ii) DiTaxa** for accurate phenotype and biomarker prediction. In both, we proposed a bootstrapping framework to investigate the sufficiency of shallow subsamples for a proper representation of the environmental sample. Sufficiency of shallow subsamples of 16S rRNA sequences for the environment or host phenotype prediction is important because (i) sub-sampling reduces the preprocessing run-time, and (ii) more importantly, it proves that even a shallow 16S rRNA sequencing (which is even more cost-effective) is enough.

MicroPheno: k-mer based host phenotype prediction

We proposed MicroPheno, a k-mer based host phenotype prediction method from 16S rRNA, which outperformed the state-of-the-art OTU features in phenotype detection (Asgari, Garakani, et al. 2018) in body-sites prediction, as well as identification of Crohn's and Rheumatoid Arthritis diseases. The k-mer representations are easy to compute at no computational cost for any type of sequence alignment as needed in OTU-picking pipelines. Although MicroPheno could outperform OTU features in phenotype prediction, short k-mers cannot be easily used as taxa distinctive biomarkers, where the OTU features are used in the state-of-the-art approach for biomarker detection (Segata, Izard, et al. 2011). This motivated me to develop an OTU-free approach for accurate biomarker detection resulted in DiTaxa.

DiTaxa: Nucleotide-pair encoding based host phenotype and biomarker detection

We proposed DiTaxa, an alignment- and reference- free, subsequence based paradigm for processing of 16S rRNA microbiome data for phenotype and biomarker detection. The main distinction of this approach from existing methods is substituting standard OTU-clustering (Robert C Edgar 2013) or sequence-level analysis (Callahan et al. 2016) by segmenting 16S rRNA reads into the most frequent variable-length subsequences of a dataset. We compared the performance of DiTaxa to the state-of-the-art methods using human-associated 16S rRNA samples for periodontal disease, rheumatoid arthritis, and inflammatory bowel diseases, as well as a synthetic benchmark dataset. We show that NPE based approach improved the state-of-the-art performance in biomarker detection over 16S rRNA data while performing competitively to the k-mer based state-of-the-art approach in phenotype prediction.

To conclude, in some use-cases in applied metagenomics only detection of the host phenotype or the environment is concerned, e.g., rapid detection of soil fertility in agriculture or criminal identification in forensic sciences. However, in some other use-cases, e.g. finding distinctive taxa for microbiome-related diseases, detection of the disease-related taxonomy is also important for diagnosis and therapy options in precision medicine. For the first scenario, we proposed MicroPheno achieving the state-of-the-art performance in phenotype detection, and for the latter, we proposed DiTaxa achieving the state-of-the-art performance in biomarker detection, while outperforming OTUs and performing competitively to the MicroPheno in the host phenotype predictions. Implementations of both MicroPheno and DiTaxa are available on the llp.berkeley.edu under Apache 2 Licenses.

Chapter 4

Data-driven processing of human languages

4.1 Introduction and chapter overview

Language technologies permeate our everyday life through the web search, translation, online shopping, email writing, spell checking systems, etc. The existence of such language technologies for a language highly depends on the existence of the underlying computational linguistic resources. Computational linguistic resources such as machine-readable lexicons, part-of-speech-taggers, and dependency parsers are available for at most a few hundred languages meaning that the majority of the 7000 languages of the world are low-resource. Many EU and US programs are designed to address this issue (Cieri et al. 2016). Multilingualism is one of the fundamental values of the global culture, which is challenged in the digital age by the gap between advances in language technologies for English versus other low-resource languages. Lack of technological support for such low-resource languages, as a result of having limited linguistic resources, reduces their use over time and puts them in danger of extinction. Even "small" languages are important for the preservation of the common heritage of humankind and cultural diversity; this can potentially benefit everybody. In addition, certain low-resource languages such as Fulani are spoken by millions and are politically and

[¶] The content of this chapter is based on the following publications:

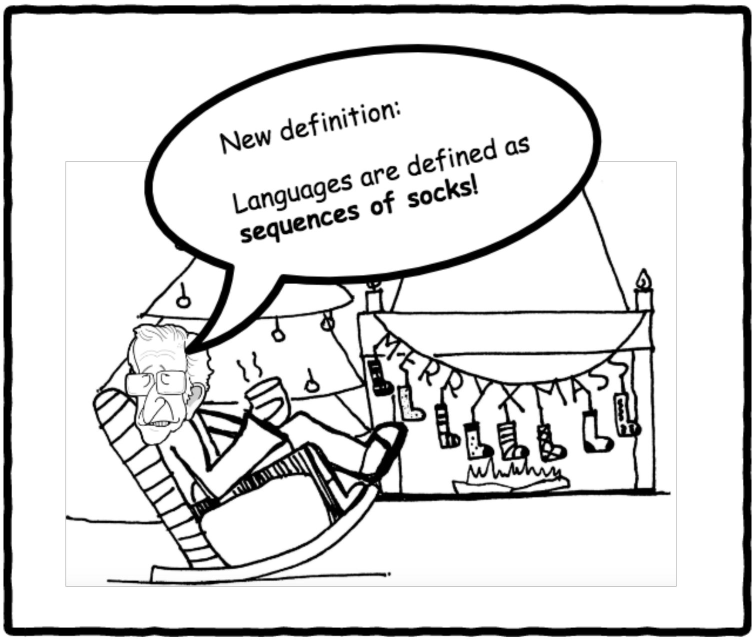
1. Asgari, E., Braune, F., Roth, B., Ringlstetter, C., & Mofrad, M. R. (2019). UniSent: Universal Adaptable Sentiment Lexica for 1000+ Languages. arXiv preprint arXiv:1904.09678.. *Under Review.*

2. Asgari, E., & Mofrad, M. R. (2016). Comparing fifty natural languages and twelve genetic languages using word embedding language divergence (weld) as a quantitative measure of language distance. *In Proceedings of the NAACL-HLT Workshop on Multilingual and Cross-lingual Methods in NLP.*

3. Asgari, E., & Schütze, H. (2017). Past, present, future: A computational investigation of the typology of tense in 1000 languages. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.*

economically important; e.g., to manage a sudden refugee crisis, natural language processing (NLP) tools would be of great benefit. Thus, providing computational linguistic resources for low-resource languages is important economically, politically, and culturally. Linguistic resources for low-resource languages either can be collected using manual annotation, which is challenging and costly and needs to be done in a collaborative framework, e.g.,[*]UniMorph Project to obtain morphological resources for even low-resource languages; or an alternative method to the manual efforts is using parallel corpora, i.e. corpora containing pairs of translated sentences. NLP has a rich literature in using language-agnostic statistical methods based on noisy channel models (Brown et al. 1993; Dyer et al. 2013) for linking words to their translations when parallel sentences are given. The linguistic annotations then from a high-resource language can be projected to the low-resource languages via such alignment links (Yarowsky et al. 2001) and generate linguistic resources for the low-resource languages semi-automatically and almost without any prior knowledge about the low-resource languages. The second approach is what we use here.

In the previous chapters on proteomics and metagenomics, we kept a language-agnostic point of view about the biological sequences. We proposed data-driven representations of DNA and protein sequences based on either their chains of monomers or their underlying probabilistic language models. In this chapter, we keep the same point of view about natural languages, assuming we have almost no linguistic knowledge about many of the world's languages, except for one or only a few of them, which is not very unrealistic in terms of available computational linguistic resources. We develop language-agnostic methods to create linguistic knowledge about low-resource languages using parallel corpora. Parallel corpora are already available for many world languages through resources, e.g., bible translations in more than 1000 languages (Mayer and Cysouw 2014).



[*] http://unimorph.org

[**] Noam Chomsky's cartoon face is taken from jjmcculough's design.

Chapter overview

As discussed in section §4.1, our main motivation in this part of the dissertation is to develop language-agnostic methods for creating linguistic knowledge and resources for low-resource languages. The contributions of this chapter in natural language processing are in three folds: (i) SuperPivot for linguistic marker detection in 1000 languages with the case study of tense, (ii) using SuperPivot we create sentiment lexicon for 1000+ languages along with a domain adaptation algorithm, and (iii) word embedding language divergence as a quantitative measure of language distance for language setting comparisons.

SuperPivot for linguistic marker detection in 1000 languages

We propose SuperPivot, a computational method for analysis of low-resource languages exist in a superparallel corpus, i.e., in a corpus that contains an order of magnitude more languages than parallel corpora currently in use. We produce analysis results for the example of "tense" for more than 1000 languages, which is to the best of our knowledge the largest cross-lingual computational study performed to date. We show that SuperPivot performs well for the cross-lingual analysis of the linguistic phenomenon of tense. We extend the existing methodology for leveraging parallel corpora for typological analysis by overcoming a limiting assumption of earlier linguistic work (Cysouw 2014). SuperPivot only requires that a linguistic feature is overtly marked in a *few* of thousands of languages as opposed to requiring that it be marked in *all* languages under investigation. This method is an instance of marker detection task-type defined in section §1.4. In addition, we investigate language family prediction using the occurrence patterns of the extracted markers for the linguistic feature of interest.

UniSent: Universal Adaptable Sentiment Lexica for 1000+ Languages

In this chapter, we introduce *UniSent* universal sentiment lexica for 1000+ languages (Asgari, Braune, et al. 2019). Sentiment lexica are vital for sentiment analysis in the absence of document-level annotations, a common scenario for low-resource languages. To the best of our knowledge, *UniSent* is the largest sentiment resource to date in terms of the number of covered languages, including many low resource ones. In this work, we use *SuperPivot* and a massively parallel Bible corpus to project sentiment information from English to other languages for sentiment analysis on Twitter data. We introduce a method called *DomDrift* to mitigate the huge domain mismatch between Bible and Twitter by a confidence weighting scheme that uses domain-specific embeddings to compare the nearest neighbors for a candidate sentiment word in the source (Bible) and target (Twitter) domain. We evaluate the quality of *UniSent* in a subset of languages for which manually created ground truth was available, Macedonian, Czech, German, Spanish, and French. We show that the quality of *UniSent* is comparable to manually created sentiment resources when it is used as the sentiment seed for the task of word sentiment prediction on top of embedding representations. Furthermore, we show that emoticon sentiments could be reliably predicted in the Twitter domain using only *UniSent* and monolingual embeddings in German, Spanish, French, and Italian.

Embedding-based quantitative comparison of languages

We introduce a new measure of distance between languages based on word embedding, called word embedding language divergence (WELD). WELD is defined as the divergence between the unified similarity distribution of embedding graphs between languages. WELD only requires monolingual embeddings and the mapping function between natural language words in different languages. Such mapping is obtained using statistical alignment over parallel corpora. Using such a measure, we perform language comparison for fifty languages and twelve genetic language variations of different organisms. Using parallel corpora and monolingual embeddings a high-level similarity of languages within the same families could be detected. Furthermore, applying the same method to organisms' genomes confirms a high-level difference in the genetic language model of humans/animals versus plants. The proposed method is a step toward defining a quantitative measure of similarity between languages, with applications in language classification, genre identification, dialect identification, and evaluation of translations with no more assumptions than the raw data and linking between the textual units, which can be obtained through statistical alignment in parallel corpora.

4.2 SuperPivot for linguistic marker detection in 1000 languages

We present SuperPivot, an analysis method for low-resource languages that occur in a superparallel corpus, i.e., in a corpus that contains an order of magnitude more languages than parallel corpora currently in use. We show that SuperPivot performs well for the crosslingual analysis of the linguistic phenomenon of tense. We produce analysis results for more than 1000 languages, conducting – to the best of our knowledge – the largest crosslingual computational study performed to date. We extend existing methodology for leveraging parallel corpora for typological analysis by overcoming a limiting assumption of earlier work: We only require that a linguistic feature is overtly marked in a *few* of thousands of languages as opposed to requiring that it be marked in *all* languages under investigation.

Linguistic Marker Projection

Significant linguistic resources such as machine-readable lexicons and part-of-speech (POS) taggers are available for at most a few hundred languages. This means that the majority of the languages of the world are low-resource. Low-resource languages like Fulani are spoken by tens of millions of people and are politically and economically important; e.g., to manage a sudden refugee crisis, NLP tools would be of great benefit. Even "small" languages are important for the preservation of the common heritage of humankind that includes natural remedies and linguistic and cultural diversity that can potentially enrich everybody. Thus, developing analysis methods for low-resource languages is one of the most important challenges of NLP today.

We address this challenge by proposing a new method for analyzing what we call *superparallel corpora*, corpora that are by an order of magnitude more parallel than corpora that have been available in NLP to date. The corpus we work with in this section is the Parallel Bible Corpus (PBC) that consists of translations of the New Testament in 1169 languages. Given that no NLP analysis tools are available for most of these 1169 languages, how can we extract the rich information that is potentially hidden in such superparallel corpora? The method we propose is based on two hypotheses.

H1 Existence of overt encoding. For any important linguistic distinction f that is frequently encoded across languages in the world, there are a few languages that encode f overtly on the surface.

H2 Overt-to-overt and overt-to-non-overt projection. For a language l that encodes f, a projection of f from the "overt languages" to l in the superparallel corpus will identify the encoding that l uses for f, both in cases in which the encoding that l uses is overt and in cases in which the encoding that l uses is non-overt. Based on these two hypotheses, our method proceeds in 5 steps.

1. Selection of a linguistic feature. We select a linguistic feature f of interest. Running example: We select past tense as feature f.

2. Heuristic search for head pivot. Through a heuristic search, we find a language l^h that contains a *head pivot* p^h that is highly correlated with the linguistic feature of interest.

Running example: “ti” in Seychelles Creole (CRS). CRS “ti” meets our requirements for a head pivot well as will be verified empirically in §4.2. First, “ti” is a surface marker: it is easily identifable through whitespace tokenization and it is not ambiguous, e.g., it does not have a second meaning apart from being a grammatical marker. Second, "ti" is a good marker for past tense in terms of both “precision” and “recall”. CRS has mandatory past tense marking (as opposed to languages in which tense marking is facultative) and "ti" is highly correlated with the general notion of past tense.

This does not mean that every clause that a linguist would regard as past tense is marked with “ti” in CRS. For example, some tense-aspect configurations that are similar to English present perfect are marked with "in" in CRS, not with "ti" (e.g., ENG “has commanded” is translated as “in ordonn”).

Our goal is not to find a head language and a head pivot that is a perfect marker of f. Such a head pivot probably does not exist; or, more precisely, linguistic features are not completely rigorously defined. In a sense, one of the significant contributions of this work is that we provide more rigorous definitions of past tense across languages; e.g., "ti" in CRS is one such rigorous definition of past tense and it automatically extends (through projection) to 1000 languages in the superparallel corpus.

3. Projection of head pivot to larger pivot set. Based on an alignment of the head language to the other languages in the superparallel corpus, we project the head pivot to all other languages and search for highly correlated surface markers, i.e., we search for additional pivots in other languages. This projection to more pivots achieves three goals. First, it makes the method more *robust*. Relying on a single pivot would result in many errors due to the inherent noisiness of linguistic data and because several components we use (e.g., alignment of the languages in the superparallel corpus) are imperfect. Second, as we discussed above, the head pivot does not necessarily have high “recall”; our example was that CRS “ti” is not applied to certain clauses that would be translated using present perfect in English. Thus, moving to a larger pivot set *increases recall*. Third, as we will see below, the pivot set can be leveraged to create a *fine-grained map of the linguistic feature*. Consider clauses referring to eventualities in the past that English speakers would render in past progressive, present perfect and simple past tense. Our hope is that the pivot set will cover these distinctions, i.e., one of the pivots marks past progressive, but not present prefect and simple past, another pivot marks present perfect, but not the other two and so on. It is beyond the scope of this dissertation to verify that we can produce such an analysis for all linguistic features, but a promising example of this type of map, including distinctions like progressive and perfective aspect, is given in §4.2.

Running example: We compute the correlation of “ti” with words in other languages and select the 100 highest correlated words as pivots. Examples of pivots we find this way are Torres Strait Creole “bin” (from English “been”) and Tzotzil “laj”. “laj” is a perfective marker, e.g., “Laj meltzaj -uk” ‘LAJ be-made subj’ means “It’s done being built” (Aissen 1987).

4. Projection of pivot set to all languages. Now that we have a large pivot set, we project the pivots to all other languages to search for linguistic devices that express the linguistic feature f. Up to this point, we have made the assumption that it is easy to segment text in all languages into pieces of a size that is not too small (individual characters of the Latin alphabet would be too small) and not too large (entire sentences as tokens would be too large). Segmentation on standard delimiters is a good approximation for the majority of languages – but not for all: it undersegments some (e.g., the polysynthetic language Inuit) and oversegments others (e.g., languages that use punctuation marks as regular characters).

For this reason, we do not employ tokenization in this step. Rather we search for character n-grams ($2 \leq n \leq 6$) to find linguistic devices that express f. This implementation of the search procedure is a limitation – there are many linguistic devices that cannot be found using it, e.g., templates in templatic morphology. We leave addressing this for future work (§4.2).

Running example: We find “-ed” for English and “-te” for German as surface features that are highly correlated with the 100 past tense pivots.

5. Linguistic analysis. The result of the previous steps is a superparallel corpus that is richly annotated with information about linguistic feature f. This structure can be exploited for *the analysis of a single language* l^i that may be the focus of a linguistic investigation. Starting with the character n-grams that were found in the step “projection of pivot set to all languages”, we can explore their use and function, e.g, for the mined n-gram “-ed” in English (assuming English is the language l^i and it is unfamiliar to us). Many of the other 1000 languages provide annotations of linguistic feature f for l^i: both the languages that are part of the pivot set (e.g., Tzotzil “laj”) and the mined n-grams in other languages that we may have some knowledge of (e.g., “-te” in German).

We can also use the structure we have generated for *typological analysis across languages* following the work of Michael Cysouw. He has pioneered a new methodology for typology ((Cysouw 2014), §4.2). We do not contribute any innovations to typology in this section, but our method is a significant advancement computationally over Cysouw’s work because we overcome many of his limiting assumptions. Most importantly, our method scales to thousands of languages as we demonstrate below whereas Cysouw worked on a few dozen.

Running example: We sketch the type of analysis that our new method makes possible in §4.2. The above steps “1. heuristic search for head pivot” and “2. projection of head pivot to larger pivot set” are based on H1: we assume the **existence of overt coding** in a subset of languages. The above steps “2. projection of head pivot to larger pivot set” and “3. projection of pivot set to all languages” are based on H2: we assume that **overt-to-overt and overt-to-non-overt projection** is possible. In the rest of the section, we will refer to the method that consists of steps 1 to 5 as *SuperPivot*: “linguistic analysis of SUPERparallel corpora using surface PIVOTs”.

We make three contributions. (i) Our basic hypotheses are H1 and H2. (H1) For an important linguistic feature, there exist a few languages that mark it overtly and easily recognizably. (H2) It is possible to project overt markers to overt and non-overt markers in other languages. Based on these two hypotheses we design SuperPivot, a new method for analyzing highly parallel corpora, and show that it performs well for the crosslingual analysis of the linguistic phenomenon of tense. (ii) Given a superparallel corpus, SuperPivot can be used for the analysis of *any low-resource language* represented in that corpus. In the supplementary material, we present results of our analysis for three tenses (past, present, future) for 1163[*] languages. An evaluation of accuracy is presented in Table 4.2. (iii) We extend Michael Cysouw's pioneering work on typological analysis using parallel corpora by overcoming several limiting factors. The most important is that Cysouw's method is only applicable if markers of the relevant linguistic feature are recognizable on the surface in *all* languages. In contrast, we only assume that markers of the relevant linguistic feature are recognizable on the surface in *a small number of* languages.

SuperPivot: Description of method

1. Selection of a linguistic feature. The linguistic feature of interest f is selected by the person who performs a SuperPivot analysis, i.e., by a linguist, NLP researcher or data scientist. Henceforth, we will refer to this person as the linguist.

In this section, $f \in F = \{\text{past}, \text{present}, \text{future}\}$.

2. Heuristic search for head pivot. There are several ways for finding the head language and the head pivot. Perhaps the linguist knows a language that has a good head pivot. Or she is a trained typologist and can find the head pivot by consulting the typological literature.

In this section, we use our knowledge of English and an alignment from English to all other languages to find head pivots. (See below for details on alignment.) We define a "query" in English and search for words that are highly correlated to the query in other languages. For future tense, the query is simply the word "will", so we search for words in other languages that are highly correlated with "will". For present tense, the query is the union of "is", "are" and "am". So we search for words in other languages that are highly correlated with the "merger" of these three words. For past tense, we POS tag the English part of PBC and merge all words tagged as past tense into one past tense word.[†] We then search for words in other languages that are highly correlated with this artificial past tense word.

As an additional constraint, we do not select the most highly correlated word as the head pivot, but the most highly correlated word in a Creole language. Our rationale is that Creole languages are more regular than other languages because they are young and have not accumulated "historical baggage" that may make computational analysis more difficult.

[*] We exclude six of the 1169 languages because they do not share enough verses with the rest.

[†] Past tense is defined as tags BED, BED*, BEDZ, BEDZ*, DOD*, VBD, DOD. We use NLTK (Bird 2006).

Table 4.1 lists the three head pivots for F.

3. Projection of head pivot to larger pivot set. We first use fast_align (Dyer et al. 2013) to align the head language to all other languages in the corpus. This alignment is on the word level.

We compute a score for each word in each language based on the number of times it is aligned to the head pivot, the number of times it is aligned to another word and the total frequencies of head pivot and word. We use χ^2 (Casella and Berger 2008) as the score throughout this section. Finally, we select the k words as pivots that have the highest association score with the head pivot.

We impose the constraint that we only select one pivot per language. So as we go down the list, we skip pivots from languages for which we already have found a pivot. We set $k = 100$ in this section. Table 4.1 gives the top 10 pivots.

4. Projection of pivot set to all languages. As discussed above, the process so far has been based on tokenization. To be able to find markers that cannot be easily detected on the surface (like "-ed" in English), we identify non-tokenization-based character n-gram features in step 4.

The immediate challenge is that without tokens, we have no alignment between the languages anymore. We could simply assume that the occurrence of a pivot has scope over the entire verse. But this is clearly inadequate, e.g., for the sentence "I arrived yesterday, I am staying today, and I will leave tomorrow", it is incorrect to say that it is marked as past tense (or future tense) in its entirety. Fortunately, the verses in the New Testament mostly have a simple structure that limits the variation in where a particular piece of content occurs in the verse. We therefore make the assumption that a particular relative position in language l_1 (e.g., the character at relative position 0.62) is aligned with the same relative position in l_2 (i.e., the character at relative position 0.62). This is likely to work for a simple example like "I arrived yesterday, I am staying today, and I will leave tomorrow" across languages.

In our analysis of errors, we found many cases where this assumption breaks down. A well-known problematic phenomenon for our method is the difference between, say, VSO and SOV languages: the first class puts the verb at the beginning, the second at the end. However, keep in mind that we accumulate evidence over $k = 100$ pivots and then compute aggregate statistics over the entire corpus. As our evaluation below shows, the "linear alignment" assumption does not seem to do much harm given the general robustness of our method.

One design element that increases robustness is that we find the two positions in each verse that are most highly (resp. least highly) correlated with the linguistic feature f. Specifically, we compute the relative position x of each pivot that occurs in the verse and apply a Gaussian filter ($\sigma = 6$ where the unit of length is the character), i.e., we set $p(x) \approx 0.066$ (0.066 is the density of a Gaussian with $\sigma = 6$ at $x = 0$) and center a bell curve around x. The total score for a position x is then the sum of the filter values at x summed over all occurring pivots. Finally, we select the positions $x_{\min}$ and $x_{\max}$ with lowest and highest values for each verse.

χ^2 is then computed based on the number of times a character n-gram occurs in a window of size w around $x_{\max}$ (positive count) and in a window of size w around $x_{\min}$ (negative count). Verses in which no pivot occurs are used for the negative count in their entirety. The top-ranked character n-grams are then output for analysis by the linguist. We set $w = 20$.

5. Linguistic analysis. We now have created a structure that contains rich information about the linguistic feature: for each verse we have relative positions of pivots that can be projected across languages. We also have maximum positions within a verse that allow us to pinpoint the most likely place in the vicinity of which linguistic feature f is marked in all languages. This structure can be used for the analysis of individual low-resource languages as well as for typological analysis. We will give an example of such an analysis in §4.2.

6. Hierarchical clusterings of markers and languages. As an additional evaluation, we worked on hierarchical clusterings of past, present and future pivots. As detailed in §4.2.4, we represent each verse by a vector of length 100 showing which pivot markers are used to express this verse. The other way of looking at these data is that for each marker we have an occurrence distribution over verses and we may exploit these data to demonstrate the distance between markers. For the purpose of comparing two markers, we propose calculation of the Jensen-Shannon divergence between the normalized occurrence distribution over verses:

$$D_{m_{p_i},m_{p_j}} = JSD(\hat{m_{p_i}}, \hat{m_{p_j}}),$$

where $\hat{m_{p_i}}$ and $\hat{m_{p_i}}$, are the normalized occurrence distributions over verses. We compare the obtained distance between markers with genetic distance of their corresponding languages using WALS information (Dryer et al. 2005). For visualization purposes, we perform Unweighted Pair Group Method with Arithmetic Mean (UPGMA) hierarchical clustering on the pairwise distance matrix of the marker for each tense separately (Johnson 1967).

In addition to clustering of pivot markers for each tense separately, we performed the same comparison for all top markers of 1107 languages* and take the average distances of languages in past, present, and future marking. This allows us to compare the average tense behavior of languages.

$$D_{l_i,l_j} = \frac{1}{3}(JSD_{past} + JSD_{present} + JSD_{future}),$$

Data, experiments and results

Data

We use a New Testament subset of the Parallel Bible Corpus (PBS) (Mayer and Cysouw 2014) that consists of 1556 translations of the the Bible in 1169 unique languages. We consider two languages to be different if they have different ISO 639-3 codes.

The translations are aligned on the verse level. However, many translations do not have complete coverage, so that most verses are not present in at least one translation. One reason for this is that sometimes several consecutive verses are merged, so that one verse contains material that is in reality not part of it and the merged verses may then be missing from the

*We exclude languages that have fewer than 7000 verses in common with the pivot language to ensure quality of marker.

		past			present			future	
	code	language	pivot	code	language	pivot	code	language	pivot
head pivots	CRS	Seychelles C.	*ti*	PAP	Papiamentu	*ta*	TPI	Tok Pisin	*bai*
	GUX	GourmanchÃ©ma	*den*	NOB	Norwegian BokmÃ¥l	*er*	LID	Nyindrou	*kameh*
	MAW	Mampruli	*daa*	HIF	Fiji Hindi	*hei*	GUL	Sea Island C.	*gwine*
	GFK	Patpatar	*ga*	AFR	Afrikaans	*is*	TGP	Tangoa	*pa*
	YAL	Yalunka	*yi*	DAN	Danish	*er*	BUK	Bugawac	*oc*
	TOH	Gitonga	*di*	SWE	Swedish	*är*	BIS	Bislama	*bambae*
	DGI	Northern Dagara	*tι*	EPO	Esperanto	*estas*	PIS	Pijin	*bae*
	BUM	Bulu (Cameroon)	*nga*	ELL	Greek	*ϵίναι*	APE	Bukiyip	*eke*
	TCS	Torres Strait C.	*bin*	HIN	Hindi	*haai*	HWC	Hawaiian C.	*goin*
	NDZ	Ndogo	*gìì*	NAQ	Khoekhoe	*ra*	NHR	Nharo	*gha*

Table 4.1. Top ten past, present, and future tense pivots extracted from 1163 languages. C. = Creole

translation. Thus, there is a trade-off between number of parallel translations and number of verses they have in common. Although some preprocessing was done by the authors of the resource, many translations are not preprocessed. For example, Japanese is not tokenized. We also observed some incorrectness and sparseness in the metadata. One example is that one Fijian translation (see §4.2) is tagged fij_hindi, but it is Fijian, not Fiji Hindi.

We use the 7958 verses with the best coverage across languages.

Experiments

1. Selection of a linguistic feature. We conduct three experiments for the linguistic features past tense, present tense and future tense.
2. Heuristic search for head pivot. We use the queries described in §4.2 for finding the following three head pivots. (i) Past tense head pivot: "ti" in Seychellois Creole (CRS) (McWhorter 2005). (ii) Present tense head pivot: "ta" in Papiamentu (PAP) (Andersen 1990). (iii) Future tense head pivot: "bai" in Tok Pisin (TPI) (Traugott 1978; Sankoff 1990).
3. Projection of head pivot to larger pivot set. Using the method described in §4.2, we project each head pivot to a set of $k = 100$ pivots. Table 4.1 gives the top 10 pivots for each tense.
4. Projection of pivot set to all languages. Using the method described in §4.2, we compute highly correlated character n-gram features, $2 \leq n \leq 6$, for all 1163 languages.
See §4.2 for the last step of SuperPivot: **5. Linguistic analysis.**

Evaluation

We rank n-gram features and retain the top 10, for each linguistic feature, for each language and for each n-gram size. We process 1556 translations. Thus, in total, we extract $1556 \times 5 \times 10$ n-grams.

Table 4.2 shows Mean Reciprocal Rank (MRR) for 10 languages. The rank for a particular ranking of n-grams is the first n-gram that is highly correlated with the relevant tense; e.g.,

language	past	present	future	all
Arabic	1.00	0.39	0.77	0.72
Chinese	0.00	0.00	0.87	0.29
English	1.00	1.00	1.00	1.00
French	1.00	1.00	1.00	1.00
German	1.00	1.00	1.00	1.00
Italian	1.00	1.00	1.00	1.00
Persian	0.77	1.00	1.00	0.92
Polish	1.00	1.00	0.58	0.86
Russian	0.90	0.50	0.62	0.67
Spanish	1.00	1.00	1.00	1.00
all	0.88	0.79	0.88	0.85

Table 4.2. MRR results for step 4. See text for details.

character subsequences of the name “Paulus” are evaluated as incorrect, the subsequence “-ed” in English as correct for past. MRR is averaged over all n-gram sizes, $2 \leq n \leq 6$. Chinese has consistent tense marking only for future, so results are poor. Russian and Polish perform poorly because their central grammatical category is aspect, not tense. The poor performance on Arabic is due to the limits of character n-gram features for a “templatic” language.

During this evaluation, we noticed a surprising amount of variation within translations of one language; e.g., top-ranked n-grams for some German translations include names like “Paulus”. We suspect that for literal translations, linear alignment (§4.2) yields good n-grams. But many translations are free, e.g., they change the sequence of clauses. This deteriorates mined n-grams. See §4.2.

Hierarchical clusterings of markers. Hierarchical clusterings of past, present and future pivots using JSD between the normalized occurrence distribution over verses are shown in Figure 4.1, Figure 4.2, and Figure 4.3 for past, present, and future tenses respectively. In addition to markers clusterings, the average tense behavior clustering of 1107 languages is depicted in Figure 4.4. In these figures languages are colored based on their language families using WALS (Dryer et al. 2005), languages without family information on WALS are uncolored. We observed that most of pivot past and future markers belong to Niger Congo family and present markers are mostly within Indo-European family. It can be seen that in many cases languages with the same family behave accordingly in tense marking. For instance, in past tense marking Oto-Manguean languages use almost the same marker of *ni* with small writing variations (Figure 4.1). Although TezoatlÃ¡n Mixtec did not have a record on WALS, since its marker is the same as other Oto-Manguean languages and works almost identical to *ni* in Oto-Manguean languages, we may guess this language is also Oto-Manguean, which turned out to be true when we performed further searches.* There were many of such cases for which we could guess the family of language based on their tense marking similarities in Figure 4.1, Figure 4.2 and Figure 4.3.

*https://www.ethnologue.com/language/mxb

	avg tense ($\frac{696}{1107}$ lang. - 103 fam.)	past ($\frac{55}{100}$ lang. - 15 fam.)	present ($\frac{70}{100}$ lang. - 17 fam.)	future ($\frac{44}{100}$ lang. - 16 fam.)
accuracy	0.93	0.55	0.81	0.58
precision	0.36	0.18	0.75	0.16
recall	0.01	0.59	0.37	0.61
TNR	0.99	0.55	0.96	0.58

Table 4.3. Language family similarity prediction results based on coordinated marking of verses. Only languages with records on WALS are included in this evaluation. TNR: true negative rate.

We use normalized JSD ($0 \leq JSD \leq 1$) for comparison of each pair of languages/markers; this allows us to investigate whether a simple threshold of 0.5 accurately predicts whether two languages are genetically related or not. The results are summarized in Table 4.3. Although the average tense marking divergence has a low recall, it expresses a high precision of 0.36, where the random chance is $\frac{1}{103} \approx 0.01$. Thus, it means that if divergence of tense marking is low the languages are very likely to be genetically related. This conclusion is supported by Figure 4.4 where many small clusters of nodes have the same color. This suggests that our method may help in completion of WALS.

A map of past tense

To illustrate the potential of our method we select five out of the 100 past tense pivots that give rise to large clusters of distinct combinations. Starting with CRS, we find other pivots that "split" the set of verses that contain the CRS past tense pivot "ti" into two parts that have about the same size. This gives us two sets. We now look for a pivot that splits one of these two sets about evenly and so on. After iterating four times, we arrive at five pivots: CRS "ti", Fijian (FIJ) "qai", Hawaiian Creole (HWC) "wen", Torres Strait Creole (TCS) "bin" and Tzotzil (TZO) "laj".

Figure 4.5 shows a t-SNE (Maaten and G. Hinton 2008) visualization of the large clusters of combinations that are found for these five languages, including one cluster of verses that do not contain any of the five pivots.

This figure is a map of past tense for all 1163 languages, not just for CRS, FIJ, HWC, TCS and TZO: once the interpretation of a particular cluster has been established based on CRS, FIJ, HWC, TCS and TZO, we can investigate this cluster in the 1164 other languages by looking at the verses that are members of this cluster. This methodology supports the empirical investigation of questions like "how is progressive past tense expressed in language X"? We just need to look up the cluster(s) that correspond to progressive past tense, look up the verses that are members and retrieve the text of these verses in language X.

To give the reader a flavor of the distinctions that are reflected in these clusters, we now list phenomena that are characteristic of verses that contain only one of the five pivots; these phenomena identify properties of one language that the other four do not have.

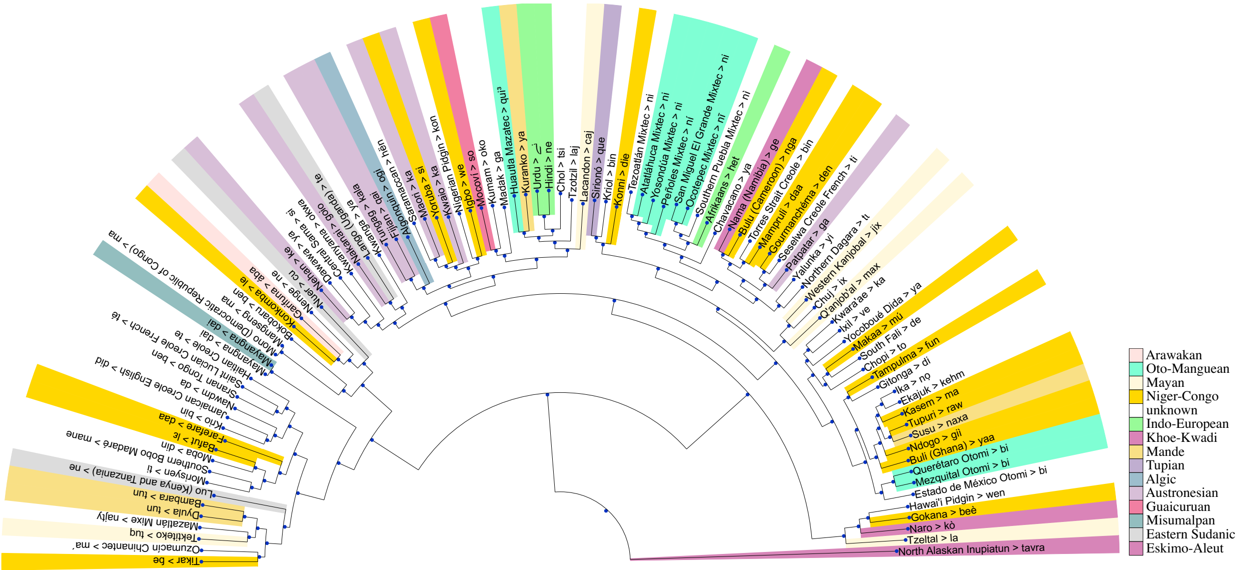


Figure 4.1. Clustering of 100 pivot past tense markers. This tree diagram illustrates the hierarchical relationships among the markers. On top of the tree structure, the colors visualize the language family of the markers. Each node is colored based on its family information. Languages with no record on WALS remained white. This clustering is based on JSD of markers in marking 5960 verses in bible. We observed that most of pivot past and future markers belong to Niger Congo family and present markers are mostly within Indo-European family. It can be seen that in many cases languages with the same family behave accordingly in tense marking. For instance, in past tense marking Oto-Manguean languages use almost the same marker of *ni* with small writing variations (Figure 4.1). Although TezoatlÃ¡n Mixtec did not have a record on WALS, since its marker is the same as other Oto-Manguean languages and works almost identical to *ni* in Oto-Manguean languages, we may guess this language is also Oto-Manguean, which turned out to be true when we performed further searches. Zooming in the electronic version is possible for a better see of the details.

CRS “ti”. CRS has a set of markers that can be systematically combined, in particular, a progressive marker “pe” that can be combined with the past tense marker “ti”. As a result, past progressive sentences in CRS are generally marked with “ti”. Example: “43004031 Meanwhile, the disciples were urging Jesus, ‘Rabbi, eat something.”’ “crs_bible 43004031 Pandan sa letan, bann disip ti pe sipliy Zezi, ‘Met! Manz en pe.”’

The other four languages do not consistently use the pivot for marking the past progressive; e.g., HWC uses “was begging” in 43004031 (instead of “wen”) and TCS uses “kip tok strongwan” ‘keep talking strongly’ in 43004031 (instead of “bin”).

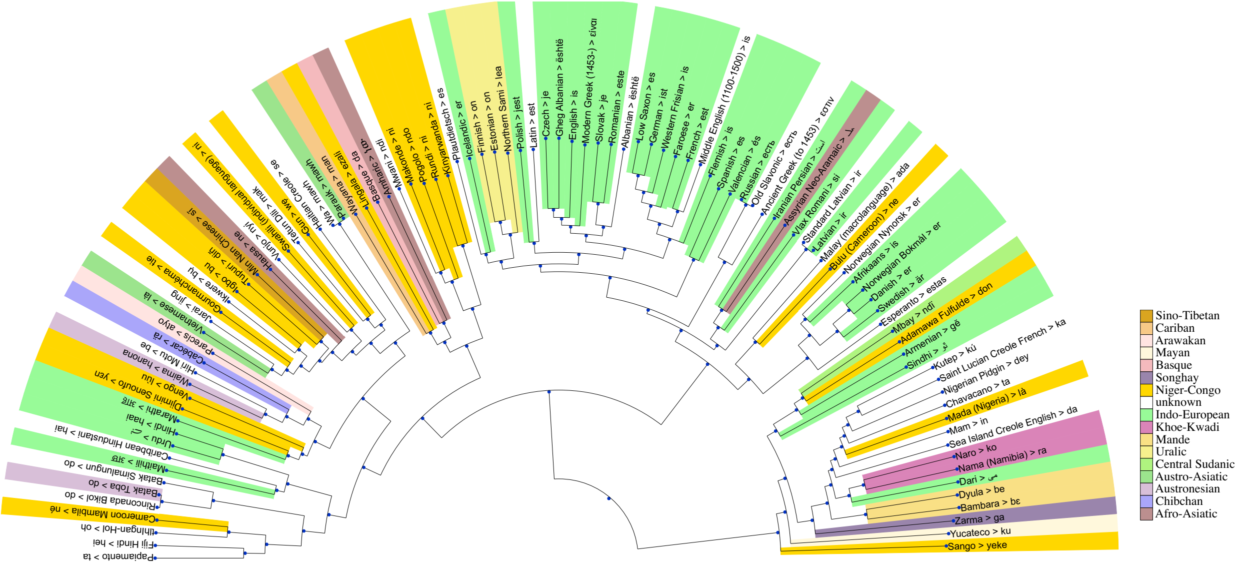


Figure 4.2. Clustering of 100 pivot present tense markers. Each node is colored based on its family information. Languages with no record on WALS remained white. This clustering is based on JSD of markers in marking 6590 verses in bible. It can be seen that in many cases languages with the same family behave accordingly in tense marking. Zooming in the electronic version is possible for a better see of the details.

FIJ "qai". This pivot means "and then". It is highly correlated with past tense in the New Testament because most sequential descriptions of events are descriptions of past events. But there are also some non-past sequences. Example: "eng_newliving 44009016 And I will show him how much he must suffer for my name's sake." "fij_hindi 44009016 Au na qai vakatakila vua na levu ni ka e na sota kaya e na vukuqu." This verse is future tense, but it continues a temporal sequence (it starts in the preceding verse) and therefore FIJ uses "qai". The pivots of the other four languages are not general markers of temporal sequentiality, so they are not used for the future.

HWC "wen". HWC is less explicit than the other four languages in some respects and more explicit in others. It is less explicit in that not all sentences in a sequence of past tense sentences need to be marked explicitly with "wen", resulting in some sentences that are indistinguishable from present tense. On the other hand, we found many cases of noun phrases in the other four languages that refer implicitly to the past, but are translated as a verb with explicit past tense marking in HWC. Examples: "hwc_2000 40026046 Da guy who wen set me up . . . " 'the guy who WEN set me up', "eng_newliving 40026046 . . . my betrayer . . . "; "hwc_2000 43008005 . . . Moses wen tell us in da Rules . . . " 'Moses WEN tell us in the rules', "eng_newliving 43008005 The law of Moses says . . . "; "hwc_2000 47006012 We wen give you guys our love . . . ", "eng_newliving 47006012 There is no lack of love on our part . . . ". In these cases, the other four languages (and English too) use a noun phrase with no

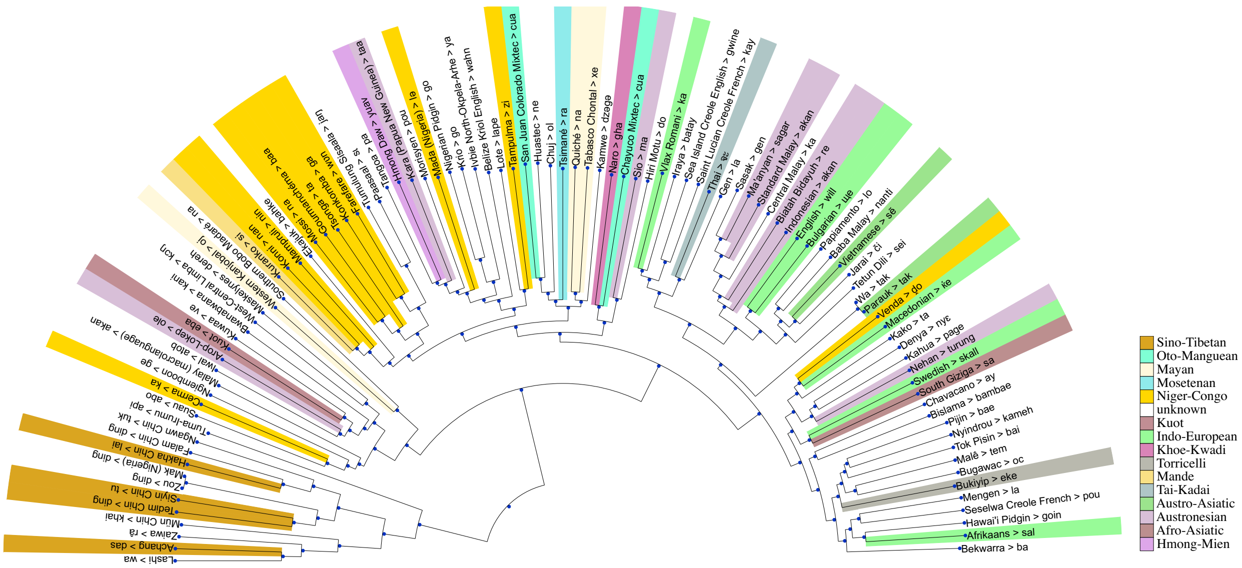


Figure 4.3. Clustering of 100 pivot future tense markers. Each node is colored based on its family information. Languages with no record on WALS remained white. This clustering is based on JSD of markers in marking 5733 verses in bible. It can be seen that in many cases languages with the same family behave accordingly in tense marking. Zooming in the electronic version is possible for a better see of the details.

tense marking that is translated as a tense-marked clause in HWC.

While preparing this analysis, we realized that HWC "wen" unfortunately does not meet one of the criteria we set out for pivots: it is not unambiguous. In addition to being a past tense marker (derived from standard English "went"), it can also be a conjunction, derived from "when". This ambiguity is the cause for some noise in the clusters marked for presence of HWC "wen" in the figure.

TCS "bin". Conditionals is one pattern we found in verses that are marked with TCS "bin", but are not marked for past tense in the other four languages. Example: "tcs_bible 46015046 Wanem i bin kam pas i da nomal bodi ane den da spiritbodi i bin kam apta." 'what came first is the normal body and then the spirit body came after', "eng_newliving 46015046 What comes first is the natural body, then the spiritual body comes later." Apparently, "bin" also has a modal aspect in TCS: generic statements that do not refer to specific events are rendered using "bin" in TCS whereas the other four languages (and also English) use the default unmarked tense, i.e., present tense.

TZO "laj". This pivot indicates perfective aspect. The other four past tense pivots are not perfective markers, so that there are verses that are marked with "laj", but not marked with the past tense pivots of the other four languages. Example: "tzo_huixtan 40010042

... ja'ch-ac'bat bendiciÃ^{3}n yu'un hech laj spas ..." (literally "a blessing ... LAJ make"), "eng_newliving 40010042 ... you will surely be rewarded." Perfective aspect and past are correlated in the real world since most events that are viewed as simple wholes are in the past. But future events can also be viewed this way as the example shows.

Similar maps for present and future tenses are presented in the Figure 4.6 and Figure 4.7.

Related work

Our work is inspired by (Cysouw 2014; Cysouw and Waelchli 2007); see also (Dahl 2007; Waelchli 2010). Cysouw creates maps like Figure 4.5 by manually identifying occurrences of the proper noun "Bible" in a parallel corpus of Jehovah's Witnesses' texts. Areas of the map correspond to semantic roles, e.g., the Bible as actor (it tells you to do something) or as object (it was printed). This is a definition of semantic roles that is complementary to and different from prior typological research because it is empirically grounded in real language use across a large number of languages. It allows typologists to investigate traditional questions from a radically new perspective.

The field of **typology** is important for both theoretical (Greenberg 1960; Whaley 1996; Croft 2002) and computational (Heiden et al. 2000; Santaholma 2007; Bender 2009; Bender 2011) linguistics. Typology is concerned with all areas of linguistics: morphology (Song 2014), syntax (Comrie 1989; Croft 2001; Croft and Poole 2008; Song 2014), semantic roles (Hartmann et al. 2014; Cysouw 2014), semantics (Koptjevskaja-Tamm et al. 2007; Dahl 2014; Waelchli and Cysouw 2012; Sharma 2009) etc. Typological information is important for many NLP tasks including discourse analysis (J. Myhill and Myhill 1992), information retrieval (Pirkola 2001), POS tagging (Bohnet and Nivre 2012), parsing (Bohnet and Nivre 2012; R. T. McDonald et al. 2013), machine translation (Hajic et al. 2000; Kunchukuttan and Bhattacharyya 2016) and morphology (Bohnet, Nivre, et al. 2013).

Tense is a central phenomenon in linguistics and the languages of the world differ greatly in whether and how they express tense (Traugott 1978; Bybee and Dahl 1989; Dahl 2000; Dahl 1985; D. Santos 2004; Dahl 2007; D. Santos 2004; Dahl 2014).

Low resource. Even resources with the widest coverage like World Atlas of Linguistic Structures (WALS) (Dryer et al. 2005) have little information for hundreds of languages. Many researchers have taken advantage of parallel information for extracting linguistic knowledge in low-resource settings (Resnik et al. 1997; Resnik 2004; Mihalcea and Simard 2005; Mayer and Cysouw 2014; Christodouloupoulos and Steedman 2015; Lison and JOerg Tiedemann 2016).

Parallel corpora and annotation projection

In general, parallel corpora are a resource of immense importance in natural language processing at least since (Brown et al. 1993)'s work on machine translation and they are widely used. In addition to machine translation, other applications include typology (Asgari and M. R. K. Mofrad 2016; Malaviya et al. 2017) and paraphrase mining (Bannard and Callison-Burch 2005).

Annotation projection is a specific use of parallel corpora: a set of labels that is available for L_1 is projected to L_2 via alignment links within the parallel corpus. L_1 labels can either be obtained through manual annotation or through an analysis module that may be available for L_1, but not for L_2. We interpret label here broadly, including, e.g., part of speech labels, morphological tags and segmentation boundaries, sense labels, mood labels, event labels, syntactic analysis and coreference. We can only cite a small subset of papers using annotation projection published in the last two decades: (T. McEnery and Richard Xiao 1999), (Ide 2000), (Yarowsky et al. 2001), (RZ Xiao and A. McEnery 2002), (Diab and Resnik 2002), (Hwa et al. 2005), (Mukerjee et al. 2006), (Pado and Lapata 2005), (Das and Petrov 2011), (Souza and Orasan 2011), (Nordrum 2015), (Marasovic and Frank 2016) and (Agic et al. 2016).

Of particular relevance is work that projects tense: (Spreyer and Frank 2008), (Xue et al. 2013), (Loaiciga et al. 2014), (Y. Zhang and Xue 2014) and (Friedrich and Gateva 2017).

In contrast to this previous work, the labels we project in this section are not the result of human annotation nor the result of the annotation computed by an NLP analysis module. Instead we interpret words in L_1 as annotation labels (words like CRS “ti” and TZO “laj”) and project these word annotation labels to another language L_2.

Discussion

Our motivation is not to develop a method that can then be applied to many other corpora. Rather, our motivation is that many of the more than 1000 languages in the Parallel Bible Corpus are low-resource and that providing a method for creating the first richly annotated corpus (through the projection of annotation we propose) for many of these languages is a significant contribution.

The original motivation for our approach is provided by the work of the typologist Michael Cysouw. He created the same type of annotation as we, but he produced it manually whereas we use automatic methods. But the structure of the annotation and its use in linguistic analysis is the same as what we provide.

The basic idea of the utility of the final outcome of SuperPivot is that the 1163 languages all richly annotate each other. As long as there are a few among the 1163 languages that have a clear marker for linguistic feature f, then this marker can be projected to all other languages to richly annotate them. For any linguistic feature, there is a good chance that a few languages clearly mark it. Of course, this small subset of languages will be different for every linguistic feature.

Thus, even for extremely resource-poor languages for which at present no annotated resources exist, SuperPivot will make available richly annotated corpora that should advance linguistic research on these languages.

Conclusion

We presented SuperPivot, an analysis method for low-resource languages that occur in a superparallel corpus, i.e., in a corpus that contains an order of magnitude more languages than parallel corpora currently in use. We showed that SuperPivot performs well for the crosslingual analysis of the linguistic phenomenon of tense. We produced analysis results for more than 1000 languages, conducting – to the best of our knowledge – the largest crosslingual computational study performed to date. We extended existing methodology for leveraging parallel corpora for typological analysis by overcoming a limiting assumption of earlier work. We only require that a linguistic feature is overtly marked in a *few* of thousands of languages as opposed to requiring that it be marked in *all* languages under investigation.

Future directions

There are at least two future directions that seem promising to us.

- Creating a common map of tense along the lines of Figure 4.5, but unifying the three tenses
- Addressing shortcomings of the way we compute alignments: (i) generalizing character n-grams to more general features, so that templates in templatic morphology, reduplication and other more complex manifestations of linguistic features can be captured; (ii) use n-gram features of different lengths to account for differences among languages, e.g., shorter ones for Chinese, longer ones for English; (iii) segmenting verses into clauses and performing alignment not on the verse level (which caused many errors in our experiments), but on the clause level instead; (iv) using global information more effectively, e.g., by extracting alignment features from automatically induced bi- or multilingual lexicons.

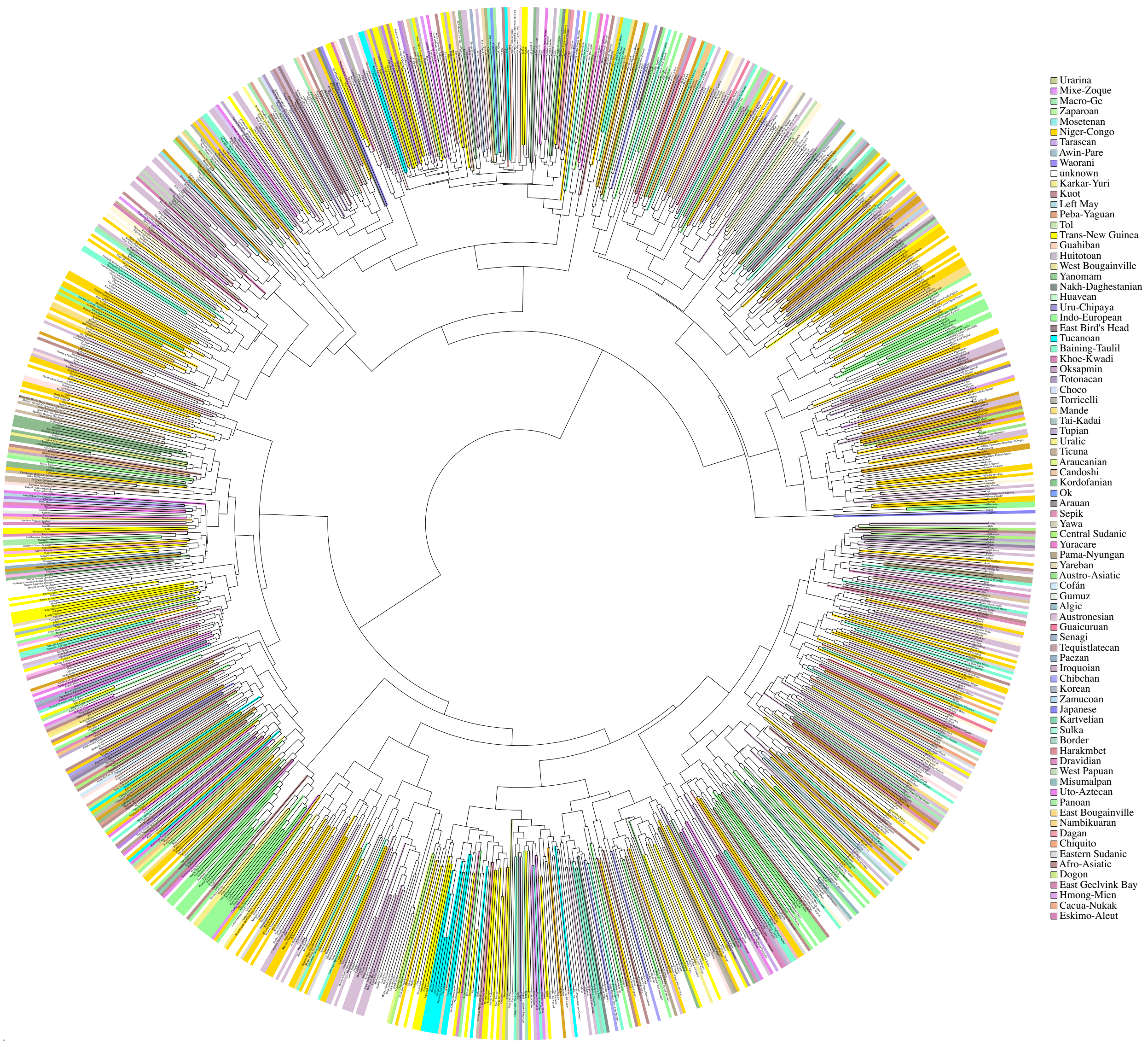


Figure 4.4. Clustering of 1107 languages based on the average Jensen-Shannon divergence in past, present, and future marking of their respective top markers. Each node is colored based on its family information. Languages with no record on WALS remained white. It can be seen that many small clusters of nodes have the same color, which together with our quantitative evaluation supports that if divergence of tense marking is low the languages are very likely to be genetically related. Zooming in the electronic version is possible for a better see of the details.

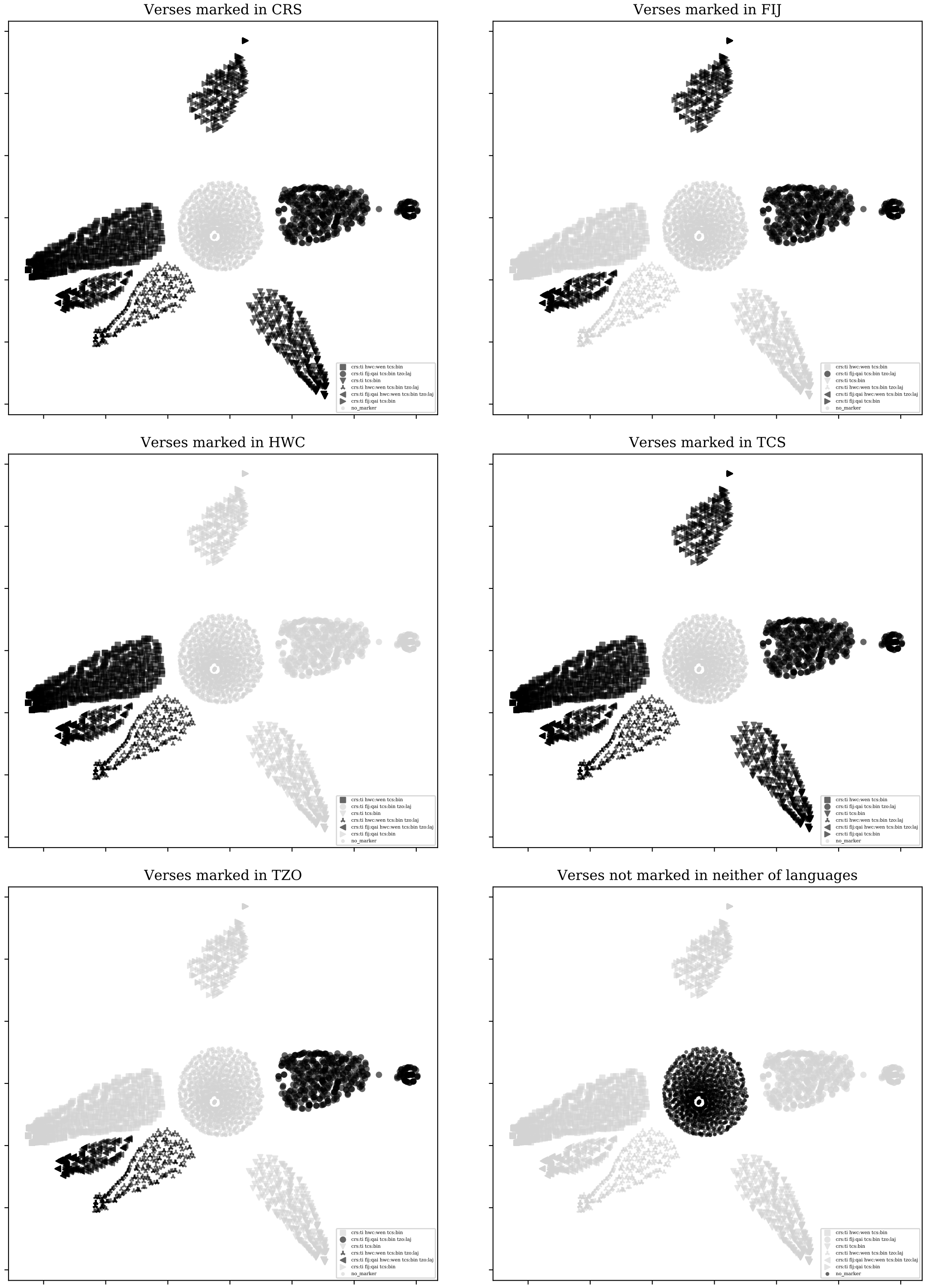


Figure 4.5. A map of past tense based on the largest clusters of verses with particular combinations of the past tense pivots from Seychellois Creole (CRS), Fijian (FIJ), Hawaiian Creole (HWC), Torres Strait Creole (TCS) and Tzotzil (TZO). For each of the five languages, we present a subfigure that highlights the subset of verse clusters that are marked by the pivot of that language. The sixth subfigure highlights verses not marked by any of the five pivots.

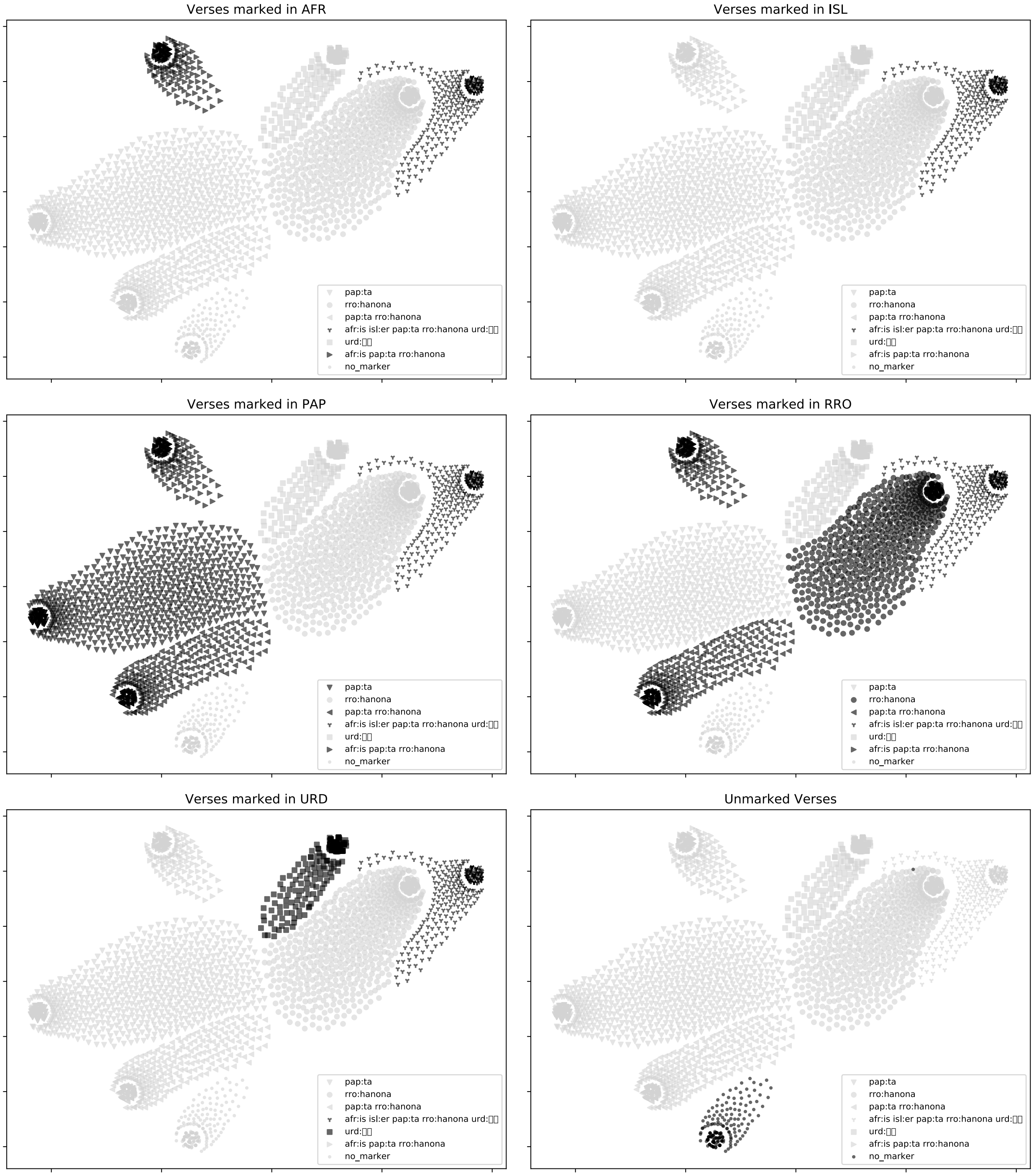


Figure 4.6. A map of present tense based on the largest clusters of verses with particular combinations of the past tense pivots from Papiamento (PAP), Waima (RRO), Afrikaans (ARF), Urdu (URD) and Icelandic (ISL). For each of the five languages, we present a subfigure that highlights the subset of verse clusters that are marked by the pivot of that language. The sixth subfigure highlights verses not marked by any of the five pivots.

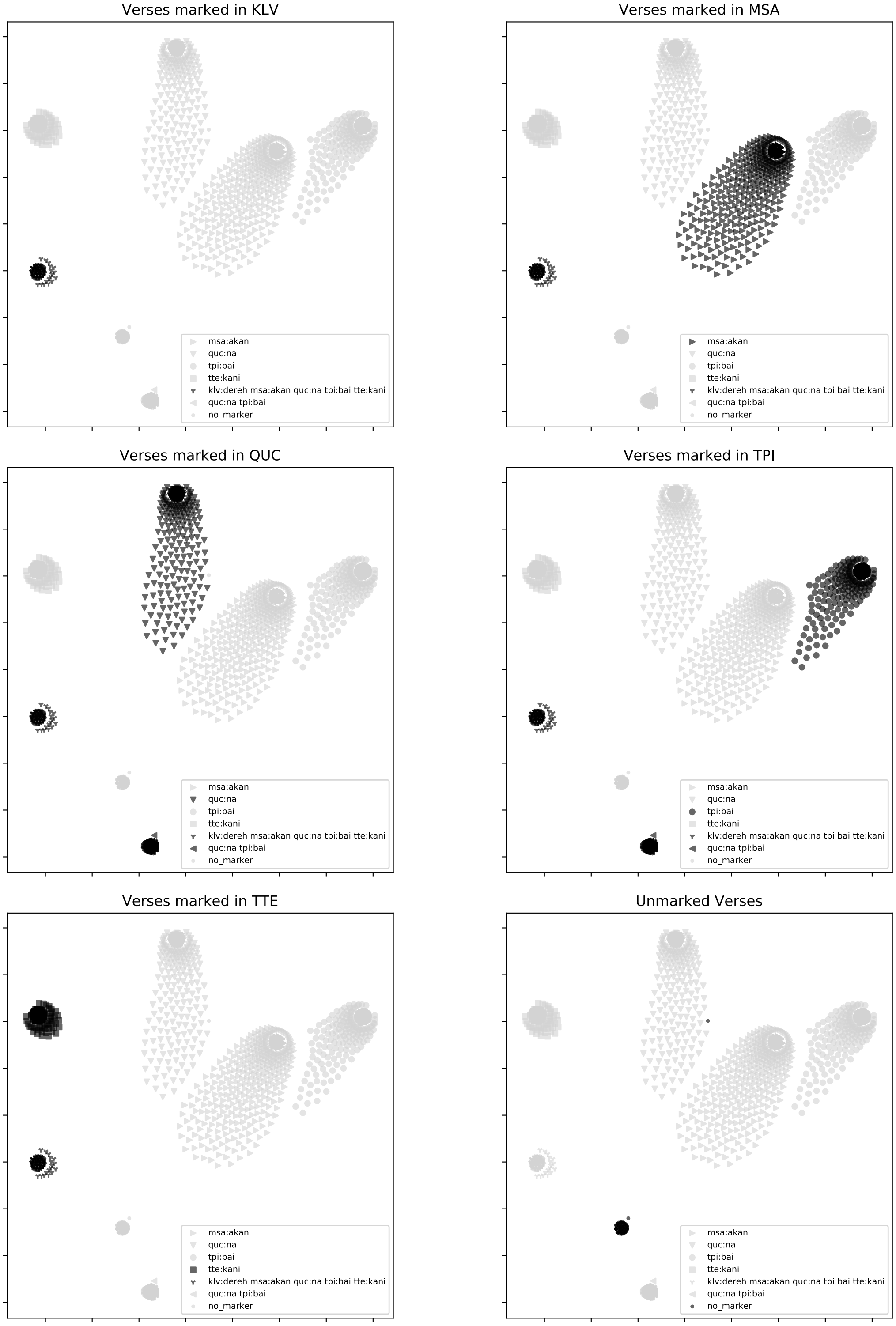


Figure 4.7. A map of future tense based on the largest clusters of verses with particular combinations of the past tense pivots from Bwanabwana (TTE), Tok Pisin (TPI), QuichÃ© (QUC), Malay (MSA) and Maskelynes (KLV). For each of the five languages, we present a subfigure that highlights the subset of verse clusters that are marked by the pivot of that language. The sixth subfigure highlights verses not marked by any of the five pivots.

4.3 UniSent: Universal Adaptable Sentiment Lexica for 1000+ Languages

Language technologies permeate our everyday life through web search, translation, online shopping, email writing, spell checking systems etc. The existence of such technologies highly depends on the existence of the underlying computational linguistic resources for a language. Computational linguistic resources such as machine-readable lexica, part-of-speech-taggers and dependency parsers are available for at most a few hundred languages. This means that the majority of the 7000 languages of the world are low-resource. This gap between advances in language technologies for English versus other languages endangers multilingualism in the digital age. Languages with lack of technological support (as a result of having limited resources) are less used over time and eventually get in danger of extinction. Many EU and US programs are designed to address this issue (Cieri et al. 2016). The rationale of these projects is that even "small" languages are important for the preservation of the common heritage of humankind and cultural diversity which benefits everybody. In addition, certain low-resource languages can be also politically and economically important.

Large amount of text available on the web and applications in marketing (Bollen et al. 2011), social science (Hopkins and King 2010), political science (H. Wang et al. 2012; Wong et al. 2016) motivates sentiment analysis of news, blogs, social networks, reviews, opinions, and recommendations. However, sentiment analysis requires either word or document level sentiment annotations. Typically, these are available only for a limited number of languages, preventing accurate sentiment classification in low resource setups. In these scenarios sentiment lexica are important game changers because in many cases end-to-end sentiment classification is not feasible due to a lack of document level annotations (Jurafsky and J. H. Martin 2014).

Recently, embedding-based approaches for supervised or semi-supervised word sentiment inference became popular allowing for lexicon vocabulary expansion and implicit domain adaptation (Rothe et al. 2016; W. L. Hamilton et al. 2016). Although using the embedding space as representation in word sentiment classification sufficiently addresses domain adaptation in many cases, it can be improved in situations where the domain shift for some words from the lexicon seeds in the source vocabulary (e.g Bible*) to the target vocabulary (e.g. Twitter) is large. For instance, in Biblical texts, the Spanish word *sensual* has the connotation of *sin* which has a negative polarity. But in the Twitter domain, the same word is associated with *sexy*, which has a positive polarity. In cases where the classifier using the embedding space fails to capture this shift, enhancing the model via mitigation of the domain mismatches is required.

Contributions: We release the first sentiment lexicon covering 1000+ languages and achieving macro-F1 over 0.75 on word sentiment prediction for most evaluated languages,

*Massively parallel corpora mainly exist in the Bible domain and for a smaller text size for the universal declaration of human rights (Emerson et al. 2014) .

meaning that we enable sentiment analysis in many low resource languages. The creation of *UniSent* requires only a sentiment lexicon in one language (e.g. English) and a small, but massively parallel corpus in a specific domain. We evaluate *UniSent* for word sentiment classification of Macedonian, Czech, German, Spanish, French against manually assigned sentiment polarities and show that its quality is comparable to the use of manually created resources, which is a great evidence that *UniSent* works well also for low-resource languages where we do not have resources for evaluation. Secondly, we evaluated UniSent w.r.t. the classification of emoticon sentiments in the Twitter domain, where macro-F1 of 0.79, 0.76, 0.74, and 0.76 were obtained for German, Italian, French, and Spanish respectively.

To ensure the usability of our lexica for any new domain, we propose *DomDrift*, a method requiring only a pretrained embedding space in the target domain, which is relatively a cheap resource to obtain. By comparing the source and target embedding graphs *DomDrift* quantifies the semantic changes of words in the sentiment lexicon in the new domain. This measure can hence be used to weight words in the sentiment lexicon for downstream supervised or semi-supervised sentiment analysis models. We show that on top of implicit domain adaptation, using target domain embeddings, the incorporation of domain drift scores improves sentiment classification for French, Spanish, and Macedonian.

Related Work

Several research efforts tackled the automatic creation of sentiment lexica for a multitude of languages, but these efforts resulted in the creation of resources for at most 136 languages (Y. Chen and Skiena 2014) or in lexicon covering a very specific low-resource language (Afli et al. 2017; Darwich et al. 2017). Moreover, these approaches heavily rely on linguistic resources, such as WordNet or fully trained machine translation systems, which limit them to the languages where these are available. An alternative to our approach in lexicon creation for sentiment is using minimal bilingual supervision (Hangya et al. 2018; Barnes et al. 2018a; Barnes et al. 2018b) to create document-level annotations for end-to-end sentiment classifications of documents. Later approaches only work in an end-to-end fashion and do not allow to directly create sentiment lexica.

The annotation projection to create *UniSent* sentiment lexica is inspired by *SuperPivot* introduced in (Asgari and Schütze 2017) for the typological analysis of tense in 1000 languages. (Agic et al. 2016) also use massively parallel corpora to project POS tags and dependency relations across languages. In contrast to these studies, here we perform parallel projection on sentiment information for resource creation and not for typological analysis. In addition, we propose a method called *DomDrift* to mitigate the huge domain mismatch between Bible and target domain via an embedding-based confidence weighting scheme.

Methods

In the next sections, we describe (i) the main resources required for *UniSent* and (ii) the steps of its creation and adaptation to new domains. The overview of these steps is also depicted in Figure 4.8.

UniSent Required Resources

Super-parallel corpus: The dataset we will work with is the Parallel Bible Corpus (PBC). PBC consists of translations of the New Testament in 1242 languages covering an order of magnitude more languages than any other parallel corpus currently in use in natural language processing research (Mayer and Cysouw 2014).

Initial sentiment seeds: We use a high-quality English sentiment lexicon called WKWSCI (Khoo and Johnkhan 2018) as a resource to be projected on other languages.

UniSent and DomDrift

Our contributions are two-fold (i) creation of *UniSent* using a cross-lingual projection of sentiment polarities, which needs to be done only once (ii) introducing *DomDrift* a novel method for adapting *UniSent* to any newly observed domain by measuring the domain-drift of words in the new domain. The first part needs a sentiment lexicon in one language (here WKWSCI for English) as well as a massively parallel corpus (PBC). For the second part, only a pre-trained embedding space in the target domain is required. In the next sections, we illustrate our method by creating *UniSent* for one example language (for better readability). The described steps are however repeated for each of the 1000+ languages composing *UniSent*. The steps are detailed next and illustrated in Figure 4.8.

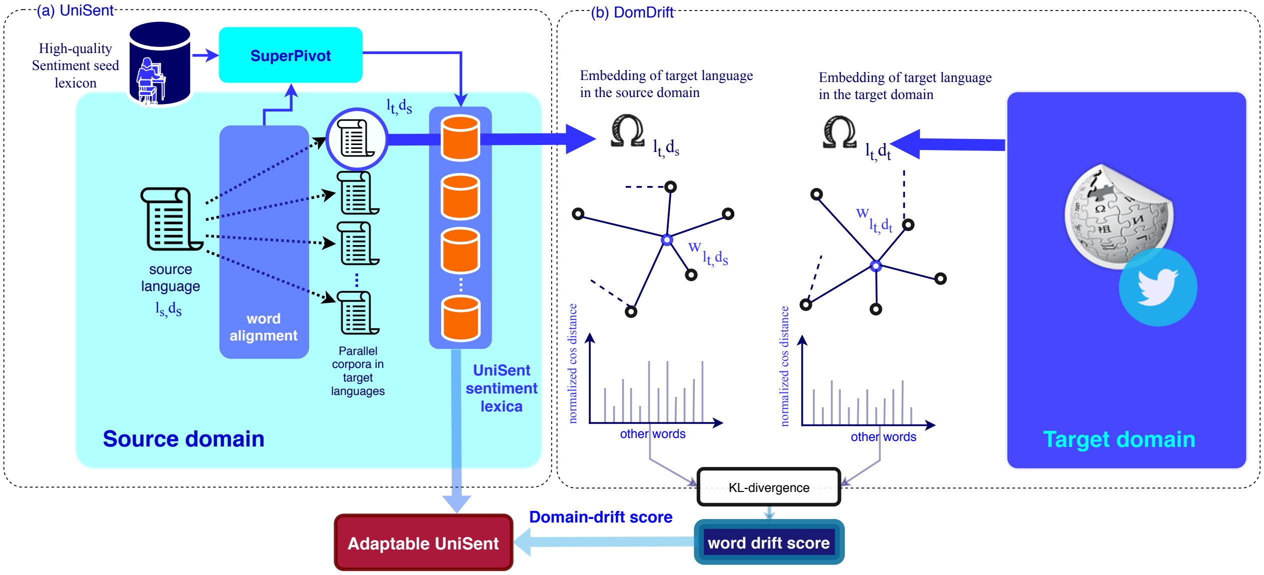


Figure 4.8. The overview of universal adaptable sentiment lexica. The approach can be divided into two main steps: (a) UniSent creation using SuperPivot method, (b) DomDrift for measurement of word domain drifts.

(i) UniSent creation using cross-lingual projection of sentiment polarities: We project sentiment polarities from English (source language) to a target language in the parallel corpus using *SuperPivot* (Asgari and Schütze 2017). This method projects annotations across 1000+ languages via an alignment graph generated using `FastAlign` (Dyer et al. 2013) on the PBC corpus. In the `FastAlign` word alignment pairs (w_{source}, w_{target}), we replace the source words with their sentiment labels from WKWSCI (where available). Subsequently, we search for the words in the target language that are highly correlated with each of the sentiment labels (positive or negative). We use FDR corrected two-sided χ^2 (Casella and Berger 2008) score to find these sentiment seeds. We denote the vocabulary of the target language in the parallel corpus by V_{l_t,d_s}, where l_t is the target language and d_s the source domain (here the domain of the parallel corpus, i.e. biblical domain).[*] This first step generates, in each target language, pairs (w_{l_t,d_s}, y), where $w_{l_t,d_s} \in V_{l_t,d_s}$ is a word in the target language and source domain and y is a highly correlated sentiment annotation with w_{l_t,d_s}. Vocabulary V_{l_t,d_s} is limited to the words in the parallel corpus [†]. Because most existing super-parallel corpora are from the Bible (Mayer and Cysouw 2014), V_{l_t,d_s} besides of being limited in size has the major drawback of originating from a very specific domain. To overcome this limitation, we define a method to measure domain drifts. In addition, we leverage word embeddings as used in (Rothe et al. 2016) to propagate the annotations y to words of a larger vocabulary than V_{l_t,d_s}. The overview of UniSent creation is depicted in Figure 4.8.a. The *UniSent* lexica and a complete list of 1242 unique languages[‡] covered by *UniSent* along with their language family information are provided in the supplementary material.

(ii) DomDrift: unsupervised measurement of domain drift: Word embeddings trained with an unsupervised language modeling objective (e.g., skip-gram) are known to preserve the syntactic and the semantic word similarities in the embedding space (Mikolov, Sutskever, K. Chen, et al. 2013; Pennington et al. 2014). Domain changes will certainly impact the neighborhoods in the embedding space. Thus a comparison of words relative distances in two embedding spaces can be used to measure their degree of domain shift (Kulkarni et al. 2015; Asgari and M. R. K. Mofrad 2016). The main purpose of *DomDrift* is to identify words in the sentiment lexicon having a domain shift in the target domain by comparison of their neighbors in the embedding spaces.

DomDrift quantifies the domain drift of a given word in the sentiment lexicon regardless of its label, only by comparison of neighbors in the source and target domains' embedding spaces $\Omega_{l_t,d_s} : V_{l_t,d_s} \rightarrow \mathbb{R}^{h_s}$, and $\Omega_{l_t,d_t} : V_{l_t,d_t} \rightarrow \mathbb{R}^{h_t}$, where h_s, h_t are the sizes of the source and target embedding spaces. *DomDrift* quantifies word domain drift as follows:

(i) In each embedding space, we compute, for each word w_{l_t,d_x} in the UniSent lexicon, the distance distribution $P(w_{l_t,d_x}, \boldsymbol{w}_{\Omega_{s,t}})$ of word w_{l_t,d_x} with all other words in intersection

[*] We call this domain *source domain* because it will be adapted to a target domain in the subsequent steps.

[†] Because we use this corpus for the cross-lingual projection

[‡] We consider two languages different if they have different ISO 639-3 codes

of source and target embedding spaces, i.e. $\forall\ w_j \in \boldsymbol{w}_{\Omega_{s,t}}$. For this, we take the $l1$ normalized cosine distance of the representation of w_{l_t,d_x} with all other words in the embedding space. This distribution can be regarded as word profile in the domain x (source or target):

$$P_i(w_{l_t,d_x}, \boldsymbol{w}_{\Omega_{s,t}}) = \frac{1 - cos(\overrightarrow{w_{l_t,d_x}}, \overrightarrow{w_i})}{\sum_j [1 - cos(\overrightarrow{w_{l_t,d_x}}, \overrightarrow{w_j})]},$$

where $w_i, w_j \in \Omega_{s,t}$ and $\overrightarrow{w_k}$ is the embedding representation of word $w_k \in \Omega_{s,t}$ in domain x (source or target). We use $\boldsymbol{w}_{\Omega_{s,t}}$ so that the word profiles in the source and target domains are comparable, i.e. they have the same elements.

(ii) We compute, for each word w_{l_t,d_x} a shift weight between vocabularies V_{l_t,d_s} and V_{l_t,d_t} of source and target spaces. This is done by comparing, for each word in the UniSent lexica, its profile in the source $P(w_{l_t,d_s}, \boldsymbol{w}_{\Omega_{s,t}})$ and target domain $P(w_{l_t,d_t}, \boldsymbol{w}_{\Omega_{s,t}})$ using the Kullback-Leibler divergence.

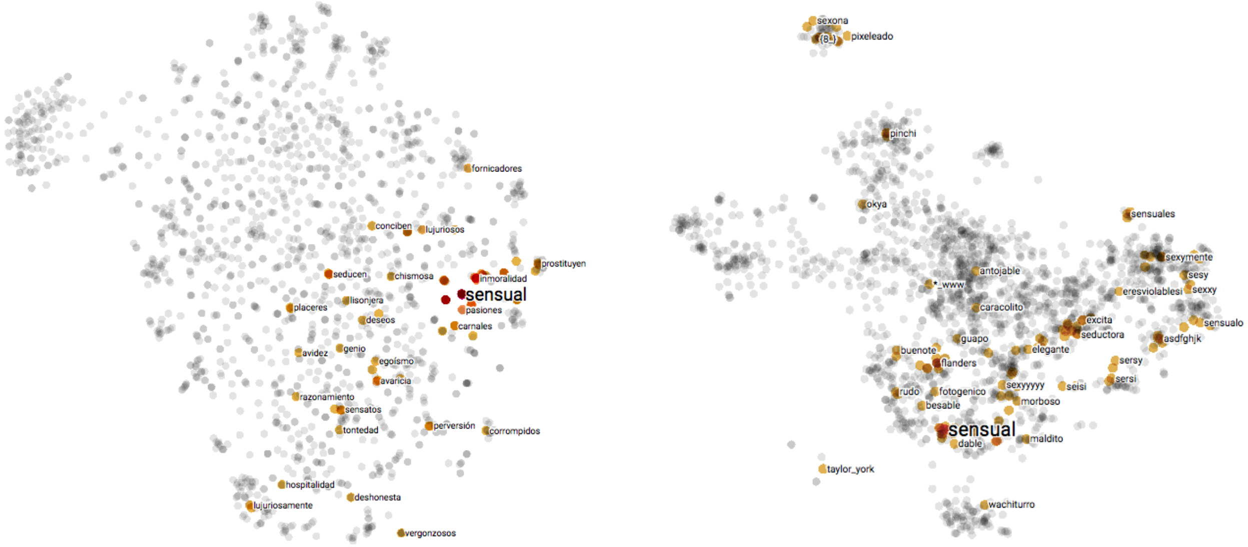
More formally for a word w' its domain drift ($\lambda_{w'}$) between the source and the target domains can be calculated as follows.

$$\lambda_{w'} = D_{\mathrm{KL}}(P(w'_{l_t,d_s}, \boldsymbol{w}_{\Omega_{s,t}}) \| P(w'_{l_t,d_t}, \boldsymbol{w}_{\Omega_{s,t}}))$$

The steps of DomDrift are illustrated in Figure 4.8.b. As also depicted in the figure, the calculated weights enhance the universal sentiment lexica resulting in a final adaptable version (Adaptable UniSent), e.g. the weights will be used to reduce the influence of huge domain mismatches in a confidence weighting scheme. In Figure 4.9 we illustrate an example of domain drift and explain the workings of our weighting method in §4.3. Once we computed our shift weights, they can be used in any semi-supervised or supervised approach (e.g., sample weights in the logistic regression model).

Source embedding Ω_{l_t,d_s}: In order to generate Ω_{l_t,d_s}, the only necessary resource is the monolingual text of PBC in the target language (source domain). For embedding creation, we use `fasttext` (Bojanowski et al. 2017) which leverages subword information within the skip-gram architecture.

Target embedding Ω_{l_t,d_t}: For generation of Ω_{l_t,d_t}, we require a monolingual text collection in the target domain to train the embedding space. An alternative is to use pretrained embeddings in the domain of interest (e.g. Twitter or News). In particular, in our experiments, we use skip-gram embeddings, pretrained on Wikipedia for French, Macedonian, Spanish, Czech, and German (Conneau et al. 2017), as well as German, Italian, French, and Spanish monolingual pretrained embeddings on Twitter, provided by (Deriu et al. 2017; Cieliebak et al. 2017).



(a) Bible embedding graph

(b) Twitter embedding graph

Figure 4.9. Neighbors of the word 'sensual' in Spanish, in the bible embedding graph (a) and the twitter embedding graph (b). Our unsupervised drift weighting method found this word in Spanish to be the most changing word from bible context to the twitter context. Looking more closely at the neighbors, the word sensual in the biblical context has been associated with negative sentiment of sins. However, in the twitter domain, it has a positive sentiment. This example shows how our unsupervised method can improve the quality of sentiment lexica.

Word sentiment classification for the evaluation of sentiment lexica

In order to evaluate *UniSent*, we use it as seed lexicon for word sentiment classification on top of embedding features. Different methods can be used to predict the sentiment of words in the target domain using embedding spaces. These include supervised methods e.g., UltraDense (Rothe et al. 2016), linear classifiers and regressors (e.g. SVM, SVR, and logistic regression) or semi-supervised methods, e.g. SentProp (W. L. Hamilton et al. 2016). In this section, we use logistic regression for the classification model. Using the language model based embedding space as the representation has two major benefits: First, the semantic continuity of the embedding space allows for propagation of sentiment labels to a larger vocabulary. This enables the annotation of further word pairs $(w_{l_t,d_t}, \hat{y})$, where $w_{l_t,d_t} \in V_{l_t,d_t}$ is a vocabulary of arbitrary size in the target domain. Secondly, using embeddings trained in a specific domain result in the implicit incorporation of semantic structures specific to the target domain (which are reflected in the embedding space), i.e. an implicit domain adaptation.

UniSent versus confidence weighted UniSent: For the word sentiment classification we train a logistic regression classifier with the annotated pairs of (w_{l_t,d_t}, y) represented in the target embedding space. In order to incorporate the drift weights, we treat the weights coming from *DomDrift* as sample weights in the logistic regression model, i.e. (w_{l_t,d_t}, y, s_w)'s are the training instances to the logistic regression classifier, where $s_w = \frac{1}{\lambda_w}$, is the seed weight calculated as the inverse of *DomDrift* score. Our evaluation in section §4.3 shows that this (simple) method is very effective and creates accurate resources. In the hyper-parameter tuning for logistic regression we also fine tune the exponent of this weight.

Experiments and Evaluation

Experimental Setup

Select Gold Standard Data As gold standard sentiment lexica for the evaluation of UniSent, we select manually created lexica in Czech (Veselovská and Ondřej Bojar 2013), German (Waltinger 2010), French (Abdaoui et al. 2014), Macedonian (Jovanoski et al. 2016), and Spanish (Perez-Rosas et al. 2012). These lexica contain general domain words (as opposed to Twitter or Bible). As gold standard for Twitter we use the emoticon dataset in (Wiebe et al. 2005; Hogenboom et al. 2013) and perform emoticon sentiment prediction for different languages.

Train-test split

In order to evaluate the UniSent, here we create train-test split for training and testing the seeds created in the projection step (see Section § 4.3). We first split *UniSent* and our gold standard lexica as illustrated in Figure 4.10. In order to design a fair evaluation, we form our training and test sets as follows:
(i) UniSent-Train-Lexicon: For the evaluation of the UniSent, we use words in *UniSent* as sentiment seeds for training in the target domain; for this purpose, we use words $w \in A \cup C$ (Figure 4.10).
(ii) Manual-Train-Lexicon: In order to obtain an upper bound for the UniSent performance, we compare the use of UniSent-Train-Lexicon against the use of words in the gold standard lexicon as sentiment seeds for the training in the target domain. For this purpose, we use words $w \in B \cup C$ (Figure 4.10).
(iii) Test-Lexicon: we randomly exclude a set of words in the {**Manual-Train-Lexicon** $\cup B$}. In the selection of the sampling size, we make sure that $UniSent - Train - Lexicon$ and $Manual - Train - Lexicon$ would contain approximately the same number of words (Figure 4.10).

Evaluation

As discussed in §4.3, we use the (manually created) English sentiment lexicon (WKWSCI) in (Khoo and Johnkhan 2018) as a resource to be projected to over 1000+ languages. We project positive and negative sentiments to create positive and negative sentiment lexica for each language.

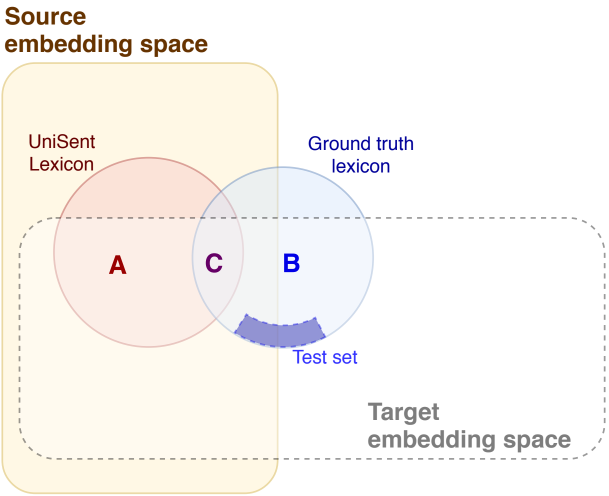


Figure 4.10. Data split used in the experimental setup of UniSent evaluation: Set (C) is the intersection of the target embedding space words (Wikipedia or Twitter) and the UniSent lexicon as well as the manually created lexicon. Set (A) is the intersection of the target embedding space words and the UniSent lexicon, excluding set (C). Set (B) is the intersection of the target embedding space words and the manually created lexicon, excluding set (C).

Our evaluation of this work is two-fold. On the one hand, we evaluate the overall quality of *UniSent* by comparing it against our manually created gold standard datasets in the Wikipedia domain. Second, we investigate the influence of *DomDrift* w.r.t. the adaptation of UniSent for Wikipedia and Twitter domains.

(i) Evaluation of UniSent vs. manually created lexica: We compare the application of Unisent for the word sentiment classification task against the manually created lexica in the following cases: (i) choice of the most frequent sentiment, (ii) use of manually created lexicon as sentiment seeds. We use the train and test seed lexica as discussed in 4.3 for training and testing of the logistic regression on top of target embedding features. We also include the sentiment classification results using the confidence weighted version of *UniSent*, where the shift between the vocabularies of the Bible and Wikipedia are calculated by *DomDrift* and are used as sample weights in the logistic regression model.

(ii) Comparison of UniSent vs. confidence weighted UniSent in the Twitter domain for emoticon prediction: To show that our adaptation method also works well on domains like Twitter, we propose a second evaluation in which we use *UniSent* together with *DomDrift* to predict the sentiment of emoticons in Twitter. Since emoticons are almost language independent, we could use the same resource for the evaluation of German, Italian, French, and Spanish, where their monolingual pretrained embeddings are available for these languages (Deriu et al. 2017; Cieliebak et al. 2017). In the adaptation step, we compute the shift between the vocabularies of Bible and Twitter. We use the *UniSent* seeds for training a logistic regression model on Twitter embedding and evaluate the classifier for the Emoticon sentiment prediction. We perform this evaluation for German, Italian, French, and Spanish, where Twitter pretrained embedding is available.

Table 4.4. We evaluate *UniSent* against the gold standard datasets in Czech, German, French, Macedonian, and Spanish. The two last columns report the accuracy and macro-F1 (averaged F1 over positive and negative classes) of *Unisent* before and after the application of the drift weighting step. The two first columns report the performance of the baseline and manually created lexicon. Note that the baseline is constantly considering the majority label.

		Manual Lexicon		UniSent Lexicon		Confidence Weighted UniSent Lexicon	
Language	Freq. sentiment baseline	target-language-specific		(projection)		(projection)	
	acc	acc	macro-F1	acc	macro-F1	acc	macro-F1
French	0.62	0.84	0.83	0.73	0.72	0.74	0.74
Macedonian	0.70	0.86	0.84	0.80	0.77	0.81	0.78
Spanish	0.64	0.82	0.80	0.78	0.76	0.80	0.77
Czech	0.62	0.87	0.87	0.82	0.81	0.79	0.78
German	0.52	0.87	0.87	0.82	0.81	0.81	0.80

Table 4.5. We evaluate *UniSent* using twitter emoticon dataset. We use monolingual Twitter embeddings in German, Italian, French, and Spanish. The two last columns report the accuracy and macro-F1 (averaged F1 over positive and negative classes) of *Unisent* before and after the application of the drift weighting step.

		UniSent Lexicon		Confidence Weighted UniSent Lexicon	
Language	Freq. sentiment baseline	(projection)		(projection)	
	acc	acc	macro-F1	acc	macro-F1
French	0.62	0.73	0.73	0.75	0.74
Spanish	0.62	0.73	0.73	0.76	0.76
German	0.62	0.80	0.79	0.80	0.79
Italian	0.62	0.76	0.76	0.75	0.75

Results

The evaluation results are reported in Tables 4.4 and 4.5. Table 4.4 compares *UniSent* and its confidence weighted version to the manually created lexica in Czech, German, French, Macedonian, and Spanish as well as a naive baseline of choosing the most frequent sentiment. For all evaluated languages, accuracy as well as macro-F1 are close to 0.8, showing that *UniSent* is a high-quality resource performing close enough to manually created seeds and clearly better than the most frequent sentiment baseline. Since in this evaluation the presented languages did not have any further advantage than having a manually created lexicon for evaluation, we can assume that UniSent would work within the same range of accuracy for any of the low-resource languages as long as a monolingual embedding (which is also cheap to obtain for low-resource languages) can be available for the target domain. Our drift weighting method brings gains in several languages: French, Macedonian, and Spanish.

In Table 4.5 we compare the quality of *UniSent* in prediction of the gold standard emoticon sentiments in the Twitter domain. The results show that (i) *UniSent* clearly outperforms the baseline of the most frequent sentiment label and (ii) our domain adaptation technique brings small improvements for French and Spanish.

In order to illustrate the function of *DomDrift* we visualized the embedding space of biblical domain and twitter domain for the word achieving the highest drift score in Spanish, i.e., word *sensual* (Figure 4.9). The neighborhood of this word in both domains is shown in the figure. In Biblical texts, this word has the connotation of *sin* which has a negative polarity. But in the Twitter domain, the same word is associated with *sexy*, which has a positive polarity. This example shows that for certain pairs of domains and languages use of *DomDrift* weights in the use of sentiment lexicon can improve the performance of sentiment analysis.

Discussion and Conclusion

In this work, we introduced *UniSent* universal sentiment lexica for 1000+ languages, which is to the best of our knowledge, the largest sentiment resource to date in terms of the number of covered languages, including many low resource ones. Although *UniSent* is created based on a specific domain (bible), our evaluation of *UniSent* on Czech, German, French, Macedonian, and Spanish showed that it can achieve macro-F1 scores $\approx$0.8 in word sentiment classification, which is comparable to the use of manually annotated resources. Given that many of covered languages in *UniSent* are very low-resource, *UniSent* can be regarded among the only computational linguistic resources available for those low-resource languages making sentiment analysis possible in such low-resource setups. In addition, through accurate prediction of Twitter emoticon sentiment for German, Italian, French, and Spanish, we showed that *UniSent* can be even used in a very different target domain and still performs quite well in sentiment analysis.

Furthermore, we proposed *DomDrift*, a method to quantify domain drift for words in the UniSent given an embedding space in the target domain. *DomDrift* compares the neighborhood of the word in the embedding spaces of source and target domains. Incorporation of *DomDrift* scores in the use of *UniSent* for sentiment classification outperformed vanilla *UniSent* in French, Spanish, and Macedonian in the Wikipedia domain and French and Spanish in the twitter domain. Not further improving the results for German, Czech and Italian languages might be because of the sufficiency of the target embedding usage for domain adaptation (Jurafsky and J. H. Martin 2014) in those. On the other hand, the fact that Spanish and French performances improved on both Wikipedia and Twitter domains when *DomDrift* is used, might show that the necessity of *DomDrift* can be related to certain property of the target language, which can be further explored as future work.

4.4 Embedding-based quantitative comparison of languages

We introduce a new measure of distance between languages based on word embedding, called word embedding language divergence (WELD). WELD is defined as divergence between unified similarity distribution of words between languages. Using such a measure, we perform language comparison for fifty natural languages and twelve genetic languages. Our natural language dataset is a collection of sentence-aligned parallel corpora from bible translations for fifty languages spanning a variety of language families. Although we use parallel corpora, which guarantees having the same content in all languages, interestingly in many cases languages within the same family cluster together. In addition to natural languages, we perform language comparison for the coding regions in the genomes of 12 different organisms (4 plants, 6 animals, and two human subjects). Our result confirms a significant high-level difference in the genetic language model of humans/animals versus plants. The proposed method is a step toward defining a quantitative measure of similarity between languages, with applications in languages classification, genre identification, dialect identification, and evaluation of translations.

Computational Comparison of Human Languages

Classification of language varieties is one of the prominent problems in linguistics (Smith 2016). The term language variety can refer to different styles, dialects, or even a distinct language (Marjorie and Rees-Miller 2001). It has been a longstanding argument that strictly quantitative methods can be applied to determine the degree of similarity or dissimilarity between languages (Kroeber and Chretien 1937; Sankaran et al. 1950; Kramsky 1959; A. McMahon and R. McMahon 2003). The methods proposed in the 1990's and early 2000' mostly relied on utilization of intensive linguistic resources. For instance, similarity between two languages was defined based on the number of common cognates or phonological patterns according to a manually extracted list (Kroeber and Chretien 1937; A. McMahon and R. McMahon 2003). Such an approach, of course, is not easily extensible to problems involving new languages. Recently, statistical methods have been proposed to automatically detect cognates (Berg-Kirkpatrick and Klein 2010; D. Hall and Klein 2010; Bouchard-Cote et al. 2013; Ciobanu and Dinu 2014) and subsequently compare languages based on the number of common cognates (Ciobanu and Dinu 2014).

In this section our aim is to define a quantitative measure of distance between languages. Such a metric should reasonably take both syntactic and semantic variability of languages into account. A measure of distance between languages can have various applications including quantitative genetic/typological language classification, styles and genres identification, and translation evaluation. In addition, comparing the biological languages generating the genome in different organisms can potentially shed light on important biological facts.

Problem Definition

Our goal is to be able to provide a quantitative estimate of distance for any two given languages. In our framework, we define a language as a weighted graph $\Omega_L(V, e)$, where V is a set of vertices (words), and $e : (V \times V) \rightarrow \Re$ is a weight function mapping a pair of words to their similarity value. Then our goal of approximating the distance between the two languages L and L' can be transferred to the approximation of the distance between $\Omega_L(V, e)$ and $\Omega_{L'}(V', e')$. In order to approach such a problem firstly we need to address the following questions:

- What is a proper weight function e estimating a similarity measure between words $w_i, w_j \in V$ in a language L?
- How can we relate words in V to words in V'?
- And finally, how can we measure a distance between languages Ω_L and $\Omega_{L'}$, which means $D(\Omega_L, \Omega_{L'})$?

In the following section we explain how researchers have addressed the above mentioned questions until now.

Word similarity within a language

The main aim of word similarity methods is to measure how similar pairs of words are to each-other, semantically and syntactically (Han et al. 2013). Such a problem has a wide range of applications in information retrieval, automatic speech recognition, word sense disambiguation, and machine translation (Collobert and Weston 2008; Glorot et al. 2011; Mikolov, Yih, et al. 2013; Turney, Pantel, et al. 2010; Resnik 1999; Schwenk 2007).

Various methods have been proposed to measure word similarity, including thesaurus and taxonomy-based approaches, data-driven methods, and hybrid techniques (Miller 1995; Mohammad and Hirst 2006; Mikolov, K. Chen, et al. 2013; Han et al. 2013). Taxonomy-based methods are not easily extensible as they usually require extensive human intervention for creation and maintenance (Han et al. 2013). One of the main advantages of data-driven methods is that they can be employed even for domains with shortage of manually annotated data.

Almost all of the data-driven methods such as matrix factorization (W. Xu et al. 2003), word embedding (Mikolov, K. Chen, et al. 2013), topic models (Blei 2012), and mutual information (Collobert, Weston, et al. 2011). The main idea in distributed representation is characterizing words by the company they keep (Geoffrey E Hinton 1984; Firth 1975; Collobert, Weston, et al. 2011).

Recently, continuous vector representations known as word vectors have become popular in natural language processing (NLP) as an efficient approach to represent semantic/syntactic units (Mikolov, K. Chen, et al. 2013; Collobert, Weston, et al. 2011). Word vectors are trained in the course of training a language model neural network from large amounts of textual data (words and their contexts) (Mikolov, K. Chen, et al. 2013). More precisely, word representations are the outputs of the last hidden layer in a trained neural network for language modeling. Thus, word vectors are supposed to encode the most relevant features to language modeling by observing various samples. In this representation similar words have closer vectors, where similarity is defined in terms of both syntax and semantics. By training word vectors over large corpora of natural languages, interesting patterns have been observed. Words with similar vector representations display multiple types of similarity. For instance, $\overrightarrow{King} - \overrightarrow{Man} + \overrightarrow{Woman}$ is the closest vector to that of the word $\overrightarrow{Queen}$ (an instance of semantic regularities) and $\overrightarrow{quick} - \overrightarrow{quickly} \approx \overrightarrow{slow} - \overrightarrow{slowly}$ (an instance of syntactic regularities). A recent work has proposed the use of word vectors to detect linguistic changes within the same language over time (Kulkarni et al. 2015). The fact that various degrees of similarity were captured by such a representation convinced us to use it as a notion of proximity for words.

Word alignment

As we discussed in section 4.4, in order to compare graphs Ω_L and Ω'_L, we need to have a unified definition of words (vertices). Thus, we need to find a mapping function from the words in V to the words in V'. Obviously when two languages have the same vocabulary set this step can be skipped, which is the case when we perform within-language genres analysis or linguistic drifts study (Stamatatos et al. 2000; Kulkarni et al. 2015), or even when we compare biological languages (DNA or protein languages) for different species (Asgari and M. R. Mofrad 2015). However, when our goal is to compare distributional similarity of words for two different languages, such as French and German, we need to find a mapping from words in French to German words.

Finding a word mapping function between two languages can be achieved using a dictionary or using statistical word alignment in parallel corpora (Franz Josef Och and Ney 2003; Lardilleux and Lepage 2009). Statistical word alignment is a vital component in any statistical machine translation pipeline (Fraser and Marcu 2007). Various methods/tools has been proposed for word alignment, such as GIZA++ (F. Och 2003) and Anymalign (Lardilleux and Lepage 2009), which are able to extract high quality word alignments from sentence-aligned multilingual parallel corpora.

One of the data resources we use in this project is a large collection of sentence-aligned parallel corpora we extract from bible translations in fifty languages. Thus, in order to find a word mapping function among all these languages we used statistical word alignment techniques and in particular Anymalign (Lardilleux and Lepage 2009), which can process any number of languages at once.

Network Analysis of Languages

The rather intuitive approach of treating languages as networks of words has been proposed and explored in the last decade by a number of researchers (Cancho and Sole 2001; HaiTao Liu and Cong 2013; Cong and Haitao Liu 2014; Gao et al. 2014). In these works, human languages, like many other aspects of human behavior, are modeled as complex networks (Costa et al. 2011), where the nodes are essentially the words of the language and the weights on the edges are calculated based on the co-occurrences of the words (HaiTao Liu and Cong 2013; Cancho and Sole 2001; Gao et al. 2014). Clustering of 14 languages based on various parameters of a complex network such as average degree, average path length, clustering coefficient, network centralization, diameter, and network heterogeneity has been done by (HaiTao Liu and Cong 2013). A similar approach is suggested by (Gao et al. 2014) for analysis of the complexity of six languages. Although, all of the above mentioned methods have presented promising results about similarity and regularity of languages, to our understanding they need the following improvements:

Measure of word similarity: Considering co-occurrences as a measure of similarity between nodes, which is the basis of the above mentioned complex network methods, is a naive estimate of similarity, (HaiTao Liu and Cong 2013; Cancho and Sole 2001; Gao et al. 2014). The most trivial cases are synonyms, which we expect to be marked as the most similar words to each other. However, since they can only be used interchangeably with each other in the same sentences, their co-occurrences rate is very low. Thus, raw co-occurrence is not necessarily a good indicator of similarity.

Independent vs. joint analysis: Previous methods have compared the parameters of language graphs independently, except for some relatively small networks of words for illustration (HaiTao Liu and Cong 2013; Cancho and Sole 2001; Gao et al. 2014). However, two languages may have similar settings of the edges but for completely different concepts. Thus, a systematic way for joint comparison of these networks is essential.

Language collection: The previous analysis was performed on a relatively small number of languages. For instance in (HaiTao Liu and Cong 2013), fourteen languages were studied where twelve of them were from the Slavic family of languages, and (Gao et al. 2014) studied six languages. Clearly, studying more languages from a broader set of language families would be more indicative.

Our Contributions

In this section, we suggest a heuristic method toward a quantitative measure of distance between languages. We propose divergence between unified similarity distribution of words as a quantitative measure of distance between languages.

Measure of word similarity: We use cosine similarity between word vectors as the metric of word similarities, which has been shown to take into account both syntactic and semantic similarities (Mikolov, K. Chen, et al. 2013). Thus, in the weighted language graph $\Omega_L(V, e)$, the weight function $e : (V \times V) \rightarrow \Re$ is defined by word-vector cosine similarities between pairs of words. Although word vectors are calculated based on co-occurrences of words within sliding windows, they are capable of attributing a reasonable degree of similarity to close words that do not co-occur.

Joint analysis of language graphs: By having word vector proximity as a measure of word similarity, we can represent each language as a joint similarity distribution of its words. Unlike the methods mentioned in section 4.4 which focused on network properties and did not consider a mapping function between nodes across various languages, we propose performing node alignment between different languages (Lardilleux and Lepage 2009). Consequently, calculation of Jensen-Shannon divergence between unified similarity distributions of the languages can provide us with a measure of distance between languages.

Language collection: In this study we perform language comparison for fifty natural languages and twelve genetic language.

Natural languages: We extracted a collection of sentence-aligned parallel corpora from bible translations for fifty languages spanning a variety of language families including Indo-European (Germanic, Italic, Slavic, Indo-Iranian), Austronesian, Sino-Tibetan, Altaic, Uralic, Afro-Asiatic, etc. This set of languages is relatively large and diverse in comparison with the corpora that have been used in previous studies (HaiTao Liu and Cong 2013; Gao et al. 2014). We calculated the Jensen-Shannon divergence between joint similarity distributions for fifty language graphs consisting of 4,097 sets of aligned words in all these fifty languages. Using the mentioned divergence we performed cluster analysis of languages. Interestingly in many cases languages within the same family clustered together. In some cases, a lower degree of divergence from the source language despite belonging to different language families was indicative of a consistent translation.

Genetic languages: Nature uses certain languages to generate biological sequences such as DNA, RNA, and proteins. Biological organisms use sophisticated languages to convey information within and between cells, much like humans adopt languages to communicate (Yandell and Majoros 2002; Searls 2002). Inspired by this conceptual analogy, we use our languages comparison method for comparison of genetic languages in different organisms. Genome refers to a sequence of nucleotides containing our genetic information. Some parts of our genome are coded in a way that can be translated to proteins (exonic regions), while some regions cannot be translated into proteins (introns) (Saxonov et al. 2000). In this study, we perform language comparison of coding regions in 12 different species (4 plants, 6 animals, and two human subjects). Our language comparison method is able to assign a reasonable relative distance between species.

Methods

As we discussed in 4.4, we transfer the problem of finding a measure of distance between languages L and L' to finding the distance between their language graphs $\Omega_L(V, e)$ and $\Omega_{L'}(V', e')$.

Word Embedding: We define the edge weight function $e : (V \times V) \rightarrow \Re$ to be the cosine similarity between word vectors.

Alignment: When two languages have different words, in order to find a mapping between the words in V and V' we can perform statistical word alignment on parallel corpora.

Divergence Calculation: Calculating Jensen-Shannon divergence between joint similarity distributions of the languages can provide us with a notion of distance between languages.

Our language comparison method has three components. Firstly, we need to learn word vectors from large amounts of data in an unsupervised manner for both of the languages we are going to compare. Secondly, we need to find a mapping function for the words and finally we need to calculate the divergence between languages. In the following section we explain each step aligned with the experiment we perform on both natural languages and genetic languages.

Learning Word Embedding

Word embedding can be trained in various frameworks (e.g. non-negative matrix factorization and neural network methods (Mikolov, Yih, et al. 2013; Levy and Goldberg 2014)). Neural network word embedding trained in the course of language modeling is shown to capture interesting syntactic and semantic regularities in the data (Mikolov, Yih, et al. 2013; Mikolov, K. Chen, et al. 2013). Such word embedding known as word vectors need to be trained from a large number of training examples, which are basically words and their corresponding contexts. In this project, in particular we use an implementation of the skip-gram neural network (Mikolov, Sutskever, K. Chen, et al. 2013).

In training word vector representations, the skip-gram neural network attempts to maximize the average probability of contexts for given words in the training data:

$$\begin{aligned} &\underset{v,v'}{\text{argmax}}\frac{1}{N}\sum_{i=1}^{N}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{i+j}|w_i) \\ &p(w_{i+j}|w_i) = \frac{\exp\left({v'}^T_{w_{i+j}} v_{w_i}\right)}{\sum_{k=1}^{W}\exp\left({v'}^T_{w_k} v_{w_i}\right)}, \end{aligned} \tag{4.1}$$

where N is the length of the training, $2c$ is the window size we consider as the context, w_i is the center of the window, W is the number of words in the dictionary and v_w and v'_w are the n-dimensional word representation and context representation of word w, respectively. At the end of the training the average of v_w and v'_w will be considered as the word vector for w. The probability $p(w_{i+j}|w_i)$ is defined using a softmax function. In the implementation we use (Word2Vec) (Mikolov, Sutskever, K. Chen, et al. 2013) negative sampling has been utilized, which is considered as the state-of-the-art for training word vector representation.

Natural Languages Data

For the purpose of language classification we need parallel corpora that are translated into a large number of languages, so that we can find the alignments using statistical methods. Recently, a massive parallel corpus based on 100 translations of the Bible has been created in XML format (Christodouloupoulos and Steedman 2015), which we choose as the database for this project. In order to make sure that we have a large enough corpus for learning word vectors, we pick the languages for which translations of both the Old Testament and the New Testament are available. From among those languages we pick the ones containing all the verses in the Hebrew version (which is the source language for most of the data) and finally we end up with almost 50 languages, containing 24,785 aligned verses. For Thai, Japanese, and Chinese we use the tokenized versions in the database (Christodouloupoulos and Steedman 2015). In addition, before feeding the skip-gram neural network we remove all punctuation.

In our experiment, we use the word2vec implementation of skip-gram (Mikolov, Sutskever, K. Chen, et al. 2013). We set the dimension of word vectors d to 100, and the window size c to 10 and we sub-sample the frequent words by the ratio $\frac{1}{10^3}$.

Genetic Languages Data

In order to compare the various genetic languages we use the IntronExon database that contains coding and non-coding regions of genomes for a number of organisms (Shepelev and Fedorov 2006). From this database we extract a data-set of coding regions (CR) from 12 organisms consisting of 4 plants (arabidopsis, populus, moss, and rice), 6 animals (sea-urchin, chicken, cow, dog, mouse, and rat), and two human subjects. The number of coding regions we have in the training data for each organism is summarized in Table 4.6. The next step is splitting each sequence to a number of words. Since the genome is composed of the four DNA nucleotides A,T,G and C, if we split the sequences in the character level the language network would be very small. We thus split each sequence into n-grams ($n = 3, 4, 5, 6$), which is a common range of n-grams in bioinformatics(Ganapathiraju et al. 2002; Mantegna et al. 1995). As suggested by(Asgari and M. R. Mofrad 2015) we split the sequence into non-overlapping n-grams, but we consider all possible ways of splitting for each sequence.

We train the word vectors for each setting of n-grams and organisms separately, again using skip-gram neural network implementation (Mikolov, Sutskever, K. Chen, et al. 2013). We set the dimension of word vectors d to 100, and window size of c to 40. In addition, we sub-sample the frequent words by the ratio 10^-3.

Word Alignment

The next step is to find a mapping between the nodes in $\Omega_L(V, e)$ and $\Omega_{L'}(V', e')$. Obviously in case of quantitative comparison of styles within the same language we do not need to find an alignment between the nodes in V and V'. However, when we are comparing two distinct

Organisms	# of CR# of 3-grams
Arabidopsis	17982442,618,288
Populus	13184428,478,304
Moss	16799938,471,771
Rice	12972634,507,116
Sea-urchin	14345727,974,115
Chicken	18776134,735,785
Cow	19646643,222,520
Dog	38114770,512,195
Mouse	21527434,874,388
Rat	19098941,635,602
Human 1	31939186,874,352
Human 2	30387277,791,232

Table 4.6. The genome data-sets used for learning word vectors in different organisms. The number of coding regions and the total occurrences of 3-grams are presented. Clearly, the total number of all n-grams (n=3,4,5,6) is almost the same.

languages we need to find a mapping from the words in language L to the words in language L'.

Word Alignment for Natural Languages

As we mentioned in section 4.4, our parallel corpora contain texts in fifty languages from a variety of language families. We decided to use statistical word alignments because we already have parallel corpora for these languages and therefore performing statistical alignment is straightforward. In addition, using statistical alignment we hope to see evidences of consistent/inconsistent translations.

We use an implementation of Anymalign (Lardilleux and Lepage 2009), which is designed to extract high quality word alignments from sentence-aligned multilingual parallel corpora. Although Anymalign is capable of performing alignments in several languages at the same time, our empirical observation was that performing alignments for all languages against a single language and then finding the global alignment through that alignment is faster and results in better alignments. We thus align all translations with the Hebrew version. To ensure the quality of alignments we apply a high threshold on the score of alignments. In a final step, we combine the results and end up with a set of 4,097 multilingual alignments. Hence we have a mapping from any of the 4,097 words in one language to one in any other given language, where the Hebrew words are unique, but not necessarily the others.

Genetic Languages Alignment

In genetic language comparison, since the n-grams are generated from the same nucleotides (A,T,C,G), no alignment is needed and V would be the same as V'.

Calculation of Language Divergence

In section 4.4 we explained how to make language graphs $\Omega_L(V, e)$ and $\Omega_{L'}(V', e')$. Then in section 4.4 we proposed a statistical alignment method to find the mapping function between the nodes in V and V'. Having achieved the mapping between the words in V and the words in V', the next step is comparison of e and e'.

In comparing language graphs what is more crucial is the *relative* similarities of words. Intuitively we know that the relative similarities of words vary in different languages due to syntactic and semantic differences. Hence, we decided to use the divergence between relative similarities of words as a heuristic measure of the distance between two languages. To do so, firstly we normalize the relative word vector similarities within each language. Then, knowing the mapping between words in V and V' we unify the coordinates of the normalized similarity distributions. Finally, we calculate the Jensen-Shannon divergence between the normalized and unified similarity distributions of two languages:

$$D_{L,L'} = JSD(\hat{e}, \hat{e'}),$$

where $\hat{e}$ and $\hat{e'}$ are normalized and unified similarity distributions of word pairs in $\Omega_L(V, e)$ and $\Omega_{L'}(V', e')$ respectively.

Natural Languages Graphs

For the purpose of language classification we need to find pairwise distances between all of the fifty languages we have in our corpora. Using the mapping function obtained from statistical alignments of Bible translations, we produce the normalized and unified similarity distributions of word pairs $\hat{e^{(k)}}$ for language $L^{(k)}$. Therefore to compute the quantitative distance between two languages $L^{(i)}$ and $L^{(j)}$ we calculate $D_{L_i,L_j} = JSD(\hat{e^{(i)}}, \hat{e^{(j)}})$.

Consequently, we calculate a quantitative distance between each pair of languages. In a final step, for visualization purposes, we perform Unweighted Pair Group Method with Arithmetic Mean (UPGMA) hierarchical clustering on the pairwise distance matrix of languages (Johnson 1967).

Genetic Languages Graphs

The same approach as carried out for natural languages is applied to genetic languages corpora. Pairwise distances of genetic languages were calculated using Jensen-Shannon divergence between normalized and unified similarity distributions of word pairs for each pair of languages.

We calculate the pairwise distance matrix of languages for each n-gram separately to verify which length of DNA segment is more discriminative between different species.

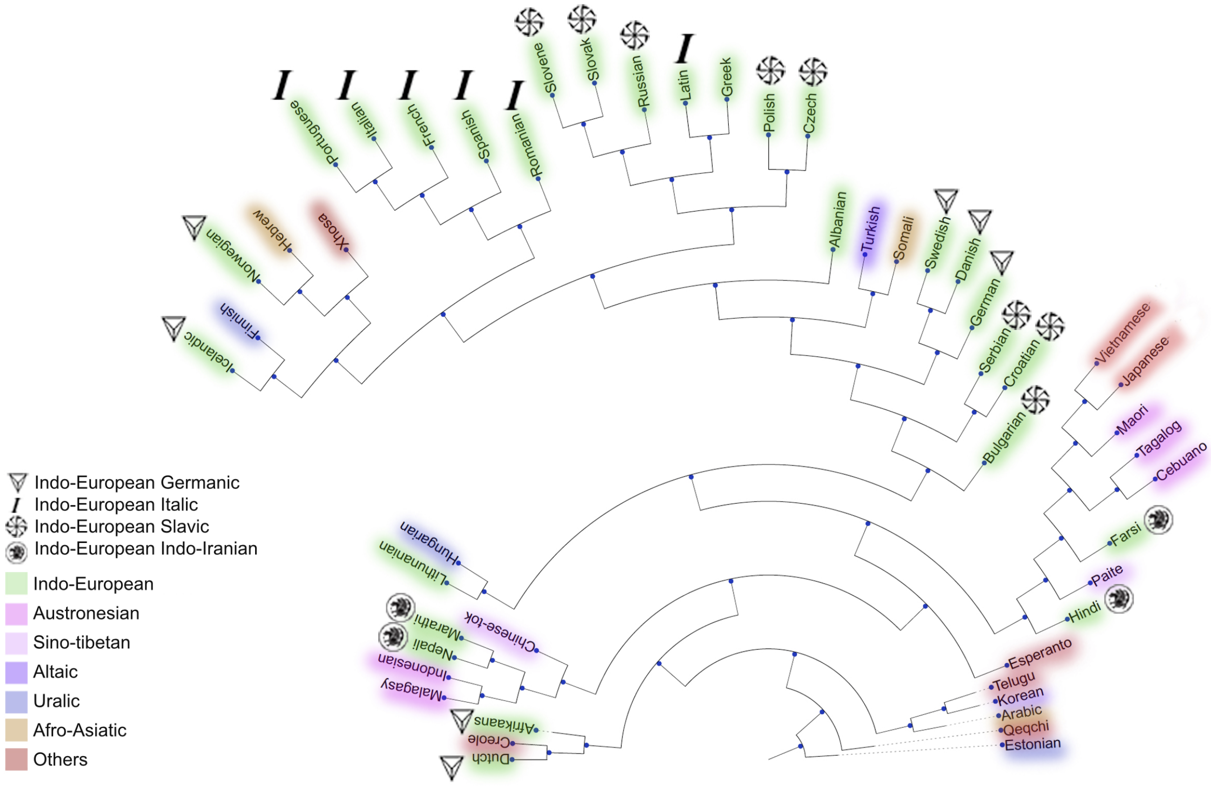


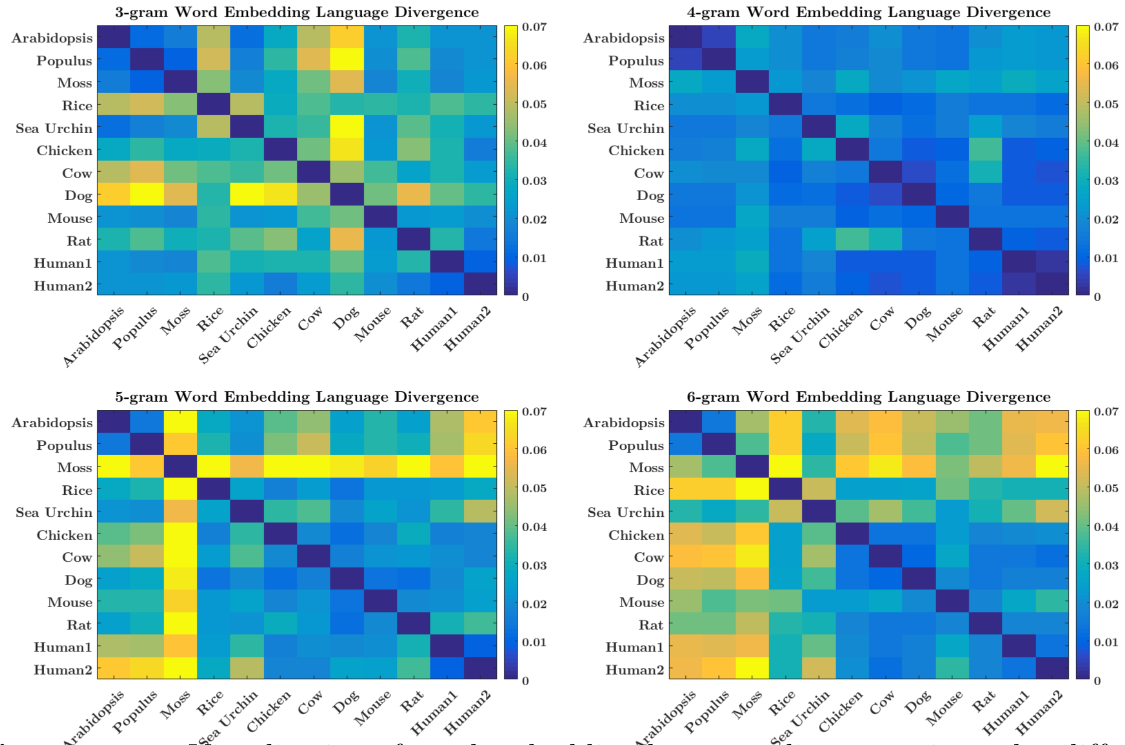
Figure 4.11. Hierarchical clustering of fifty natural languages according to divergence of joint distance distribution of 4097 aligned words in bible parallel corpora. Subsequently we use colors to show the ground-truth about family of languages. For Indo-European languages we use different symbols to distinguish various sub-families of Indo-European languages. We observe that the obtained clustering reasonably discriminates between various families and subfamilies.

Results

Classification of Natural Languages

The result of the UPGMA hierarchical clustering of languages is shown in Figure 4.11. As shown in this figure, many languages are clustered together according to their family and sub-family. Many Indo-European languages (shown in green) and Austronesian languages (shown in pink) are within a close proximity. Even the proximity between languages within a sub-family are preserved with our measure of language distance. For instance, Romanian, Spanish, French, Italian, and Portuguese, all of which belong to the Italic sub-family of Indo-European languages, are in the same cluster. Similarly, the Austronesian langauges Cebuano, Tagalog, and Maori as well as Malagasy and Indonesian are grouped together.

Although the clustering based on word embedding language divergence matches the genetic/typological classification of languages in many cases, for some pairs of languages their distance in the clustering does not make any genetic or topological sense. For instance, we expected Arabic and Somali as Afro-Asiatic languages to be within a close proximity with Hebrew. However, Hebrew is matched with Norwegian, a Germanic Indo-European


3-gram Word Embedding Language Divergence

4-gram Word Embedding Language Divergence

5-gram Word Embedding Language Divergence

6-gram Word Embedding Language Divergence

Figure 4.12. Visualization of word embedding language divergence in twelve different genomes belonging to 12 organisms for various n-gram segments. Our results indicate that evolutionarily closer species have higher proximity in the syntax and semantics of their genomes.

language. After further investigations and comparing word neighbors for several cases in these languages, it turns out that the Norwegian bible translation highly matches Hebrew because of being a consistent and high-quality translation. In this translation, synonym were not used interchangeably and language usage stays more faithful to the structure of the Hebrew text.

Divergence between Genetic Languages

The pairwise distance matrix of the twelve genetic languages for n-grams ($n = 3, 4, 5, 6$) is shown in Figure 4.12. Our results confirm that evolutionarily closer species have a reasonably higher level of proximity in their language models. We can observe in Figure 4.12, that as we increase the number of n-grams the distinction between animal/human genome and plant genome increases.

Conclusion

In this section, we proposed Word Embedding Language Divergence (WELD) as a new heuristic measure of distance between languages. Consequently we performed language comparison for fifty natural languages and twelve genetic languages. Our natural language dataset was a collection of sentence-aligned parallel corpora from bible translations for fifty languages spanning a variety of language families. We calculated our word embedding language divergence for 4,097 sets of aligned words in all these fifty languages. Using the mentioned divergence we performed cluster analysis of languages.

The corpora for all of the languages but one consisted of translated text instead of original text in those languages. This means many of the potential relations between words such as collocations and culturally influenced semantic connotations did not have the full chance to contribute to the measured language distances. This can potentially make it harder for the algorithm to detect related languages. In spite of this, however in many cases languages within the same family/sub-family clustered together. In some cases, a lower degree of divergence from the source language despite belonging to different language families was indicative of a consistent translation. This suggests that this method can be a step toward defining a quantitative measure of similarity between languages, with applications in languages classification, genres identification, dialect identification, and evaluation of translations.

In addition to the natural language data-set, we performed language comparison of n-grams in coding regions of the genome in 12 different species (4 plants, 6 animals, and two human subjects). Our language comparison method confirmed that evolutionarily closer species are closer in terms of genetic language models. Interestingly, as we increase the number of n-grams the distinction between genetic language in animals/human versus plants increases. This can be regarded as indicative of a high-level diversity between the genetic languages in plants versus animals.

4.5 Summary of contributions in human language processing

In this chapter, we introduced three computational frameworks for language-agnostic processing of human languages toward the goal of linguistic knowledge or resource creation for low-resource languages. Firstly, we introduced *SuperPivot* for linguistic marking detection in parallel settings and the typological analysis of tense in 1000+ languages. Secondly, *SuperPivot* we created a universal sentiment lexicon of 1000+ languages, which is to the best of our knowledge, the largest sentiment resource to date in terms of the number of covered languages, including many low resource ones. We proposed *DomDrift*, a method to quantify domain drift for words in the UniSent given an embedding space in the target domain, which makes UniSent adaptable to any target domain. Thirdly, we introduced a quantitative comparison of languages and language variations based on the underlying probabilistic language model reduced to language embedding graphs.

We introduced SuperPivot for subsequence based linguistic marker detection in more than 1000 languages without making any strong-assumption about the studied languages. The basic idea of SuperPivot SuperPivot is that the parallel corpora of 1000 languages, all richly annotate each other. As long as there are a few among the 1000 languages that have a clear marker for linguistic feature f, then this marker can be projected to all other languages to richly annotate them. For any linguistic feature, there is a good chance that a few languages clearly mark it. Of course, this small subset of languages will be different for every linguistic feature. However, the motivation of SuperPivot is not to develop a method that can then be applied to many other corpora. Rather, the motivation is that many of the more than 1000 languages in the Parallel Bible Corpus are low-resource and that providing a method for creating the first richly annotated corpus (through the projection of annotation we propose) for many of these languages is a significant contribution. Even for extremely resource-poor languages for which at present no annotated resources exist, SuperPivot will make available richly annotated corpora that should advance linguistic research on these languages.

We introduced *UniSent* universal sentiment lexica for 1000+ languages. Although *UniSent* is created based on a specific domain (bible), our evaluation of *UniSent* on Czech, German, French, Macedonian, and Spanish showed that it can achieve macro-F1 scores ≈0.8 in word sentiment classification, which is comparable to the use of manually annotated resources. Given that many of covered languages in *UniSent* are very low-resource, *UniSent* can be regarded among the only computational linguistic resources available for those low-resource languages making sentiment analysis possible in such low-resource setups. Furthermore, through accurate prediction of Twitter emoticon sentiment for German, Italian, French, and Spanish, we showed that *UniSent* can be even used in a very different target domain and still performs quite well in sentiment analysis. Next, we proposed *DomDrift*, a method to quantify domain drift for words in the UniSent given an embedding space in the target domain. *DomDrift* compares the neighborhood of the word in the embedding spaces of source and target domains. Incorporation of *DomDrift* scores in the use of *UniSent* for sentiment classification outperformed vanilla *UniSent* in French, Spanish, and Macedonian in the Wikipedia domain and French and Spanish in the twitter domain.

We proposed word embedding language divergence, as a new method for comparison of languages based on their word embedding graphs. Word embedding language divergence can be regarded as an extension of *DomDrift* comparing the neighborhood of all words in the embedding spaces of two languages simultaneously. Since this method is language-agnostic it can be used on any type of languages, even biological sequences. We showed that applying this method using monolingual embedding and statistical alignments on parallel corpora can be used for estimation of the language phylogenetic tree. On genome sequences since the lexical units are the same in the language variations, only monolingual embeddings are sufficient to quantify their distances. This method is an alternative method to the alignment based comparison of genomes, where high-level similarity of genomes is compared with each other instead of their point-wise differences (mutations).

Chapter 5

Conclusions and future work

In this dissertation, titled “Life Language Processing”, I investigated deep learning-based and language-agnostic processing of proteomics, genomics, metagenomics, and human languages through five different task types (where applicable): (i) sequence representations and representation learning, (ii) sequence classifications, (iii) sequence labeling, (iv) biomarker and linguistic marker detection and analysis, and (v) quantitative comparison of languages. The same insight and spirit of methods developed for three central problems in proteomics, genomics & metagenomics, and language processing:

1. **Proteomics**: How to efficiently use a large amount of existing protein primary sequences to achieve a better performance in the structural and functional annotation of protein sequences and to reduce the gap between the number of known protein sequences and known protein tertiary structures and functions?

2. **Genomics/metagenomics**: How to detect the host phenotype and the phenotype-specific taxa from the microbial samples? (which has applications in the establishment of diagnosis and therapy options in precision medicine, forensic sciences, and agriculture.)

3. **Human languages**: How to create linguistic knowledge and linguistic resources automatically for low-resource languages with zero/minimal language-dependent assumptions?

Next, I conclude the dissertation with the summary of work in the areas of **Proteomics**, **Genomics/metagenomics**, and **Human languages**. In each part, the future directions will be also discussed. Figure 5.1 summarizes the task-types and applications covered in this dissertation in these three areas.

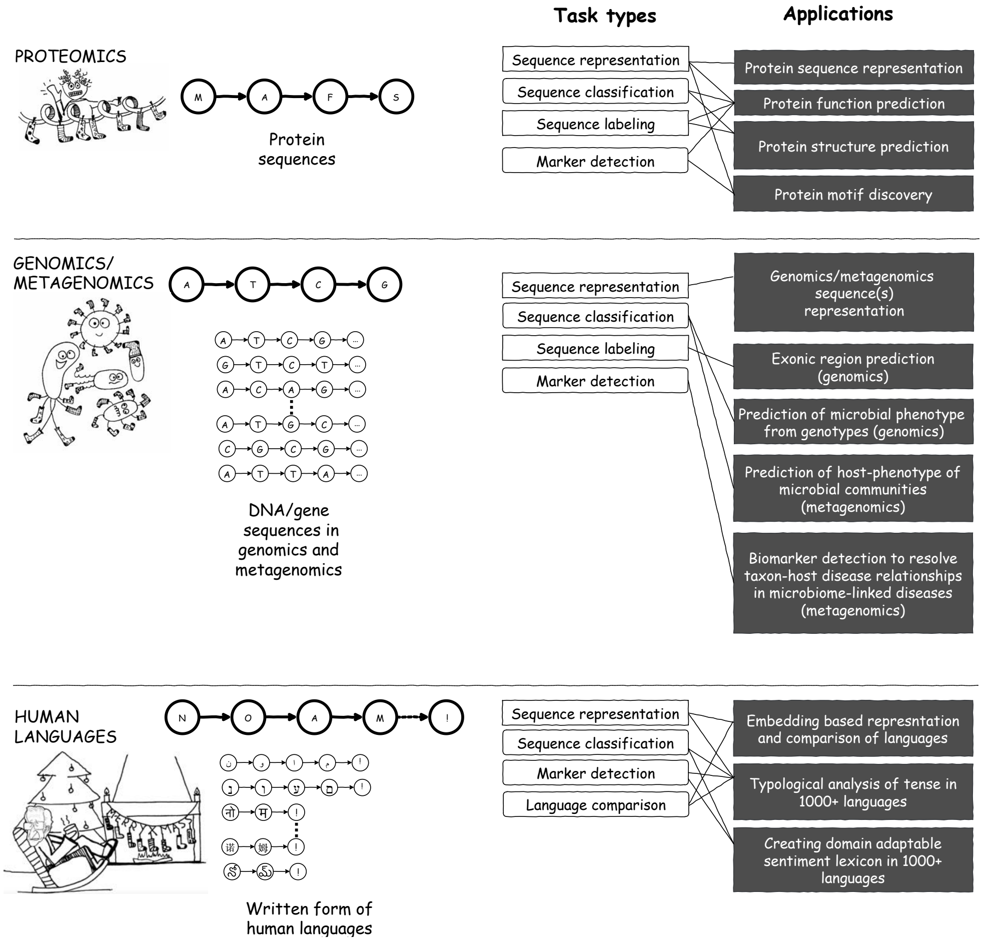

Task types
Applications
PROTEOMICS
M
A
F
S
Protein sequences
Sequence representation
Sequence classification
Sequence labeling
Marker detection
Protein sequence representation
Protein function prediction
Protein structure prediction
Protein motif discovery
GENOMICS/ METAGENOMICS
A
T
C
G
DNA/gene sequences in genomics and metagenomics
Sequence representation
Sequence classification
Sequence labeling
Marker detection
Genomics/metagenomics sequence(s) representation
Exonic region prediction (genomics)
Prediction of microbial phenotype from genotypes (genomics)
Prediction of host-phenotype of microbial communities (metagenomics)
Biomarker detection to resolve taxon-host disease relationships in microbiome-linked diseases (metagenomics)
HUMAN LANGUAGES
N
O
A
M
!
Written form of human languages
Sequence representation
Sequence classification
Marker detection
Language comparison
Embedding based represntation and comparison of languages
Typological analysis of tense in 1000+ languages
Creating domain adaptable sentiment lexicon in 1000+ languages

Figure 5.1. The contributions of life language processing project in proteomics, genomics/metagenomics, and human languages are presented. Different language processing task-types are specified for each area. The task-types are connected to the specific applications covered in this thesis.

1. Summary of life language processing for proteomics and the future directions

A word-cloud summary of proteomics-related contributions is shown in Figure 5.2. In proteomics, for the first time, we introduced a language-model based representation of protein sequences over sequence k-mers, called protein-vector or ProtVec. ProtVec representation, which is inspired by word embedding in NLP (Mikolov, Sutskever, K. Chen, et al. 2013), can be efficiently trained over large protein sequence datasets (Swiss-Prot) once and then be used as feature representation in down-stream machine learning tasks. We intrinsically evaluated ProtVec by measuring the continuity of this representation with respect to different biophysical properties and showed that this space preserves the biophysical similarity of protein k-mers.

For extrinsic evaluations, we showed that ProtVec and its variable-length extension, ProtVecX, could boost the performance of learning algorithms in protein informatics for the ultimate goal of functional and structural annotation of protein sequences, which is one the key open question in the molecular biology. In particular, we evaluated protein vectors for different protein functional and structural annotation tasks including protein family classification, disordered protein prediction, protein domain identification, secondary structure prediction, sub-cellular location prediction, identification of toxin proteins, and enzyme prediction, where combination of k-mers and protein vectors showed the most success in such machine learning predictions. Combination of embeddings and k-mers as input features to the multilayer perceptron neural networks brought us the first and the third places in two out of three protein classification tasks in the Critical Assessment of protein Function Annotation (CAFA) in 2018 (CAFA 3.14) (N. Zhou et al. 2019). Recently, ProtVec has been used and extended for variety of tasks in bioinformatics, where we only cite a subset of such papers (Wan and J. Zeng 2016; Islam et al. 2017; Hamid and Friedberg 2018; K. K. Yang et al. 2018; Jaeger et al. 2018; Du et al. 2018; A. Dutta et al. 2018; Y. Xu et al. 2018; Öztürk et al. 2018). In addition to ProtVec's contributions in bioinformatics, similar approaches afterwards were introduced for subword embedding in NLP inspired by ProtVec work (Schütze et al. 2016).

In addition, we introduced general purpose probabilistic variable length segmentation of protein sequences to replace k-mer based representation of protein sequences into unsupervised peptide-pair encoding units. This representation can be widely used in protein informatics. In particular, we proposed it for protein motif mining and the above-mentioned extension of ProtVec (ProtVecX). We presented DiMotif as an alignment-free discriminative motif mining method evaluated for finding protein motifs. The significant motifs extracted could reliably detect the integrins, integrin-binding, and biofilm formation-related proteins on a reserved set of sequences with high F1 scores. In addition, DiMotif could detect experimentally verified motifs related to nuclear localization signals with a reasonable overlap.

Furthermore, we investigated the deep learning-based protein secondary structure prediction approaches from the protein primary sequence. We focused on finding an optimal representation and deep learning predictive model for this task. The most challenging dataset for this task to-date is Q8 (8 classes) on CullPDB/CB513 dataset, where the dissimilarity of training and test set is ensured. We investigated (i) different protein sequence representations including one-hot vectors, biophysical features, protein sequence embedding (ProtVec), deep amino acid contextualized embedding (ELMo), and the Position Specific Scoring Matrix (PSSM), (ii) different deep-learning architectures including convolutional neural networks (CNN), recurrent neural networks (in particular Bi-LSTM), use of highway connection, attention mechanism, and multi-scale CNN (Jiyun Zhou et al. 2018). We showed that PSSM and its combination with one-hot vectors achieve the best performance in protein secondary structure prediction. The best performing model was the CNN-BiLSTM architecture, which captures both local and global sequence features essential for proteins secondary structure. Moreover, we performed error analysis on the most accurate model based on the location of misclassified amino acids as well as the confusion matrix analysis. Interestingly, misclassified secondary structures were significantly correlated with locating at the structural transitions.

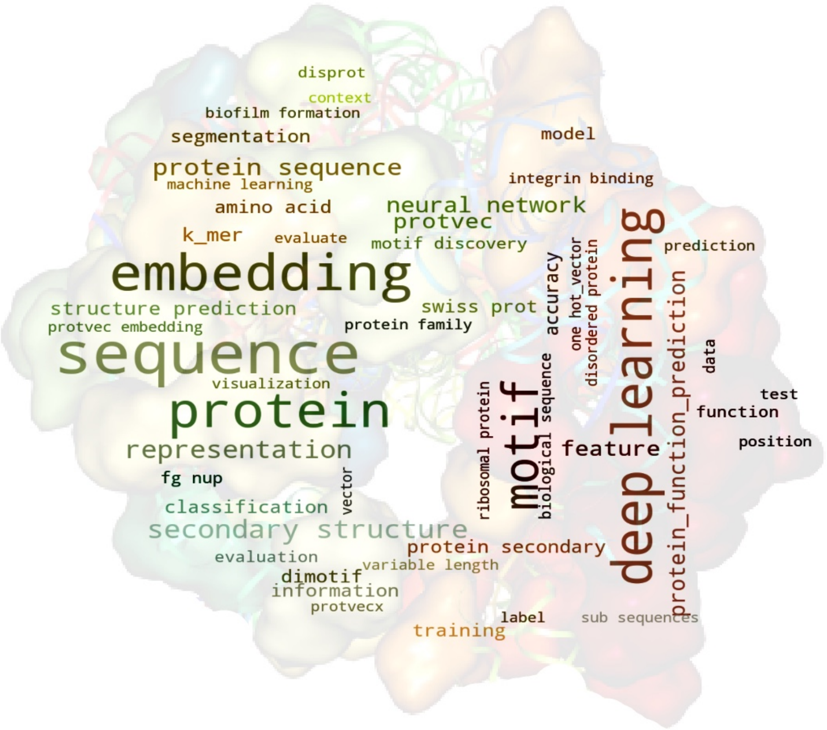


Figure 5.2. Semi-automatically generated word-cloud of the proteomics-related contributions of the dissertation.
The background images is taken from
http://blog.bitsathy.ac.in/ribosomes-the-protein-makers-now-produces-polymers.

Such a correlation is most likely due to the inaccurate assignment of the secondary structure at the boundaries in ground-truth (Y. Yang et al. 2016). By ignoring the boundary amino acids from the evaluation, the Q8 accuracy would increase for an extra %20, i.e., %90.3. Analysis of the confusion matrix furthermore indicates that similar secondary structures are highly confusing (helices: H and G as well as unstructured regions: S, T, and L) showing that the model can learn high-level information about the secondary structures.

Future directions of life language processing in proteomics:

Functional annotation: The use of protein embeddings for functional and structural annotations of sequences may be expanded to a more broad range of protein functions (e.g., the hierarchy of labels in the whole gene ontology (G. O. Consortium 2018)) and a side outcome would be a supervised tuned representation of proteins for the functional annotation. A systematic comparison of different deep neural network in the protein functional annotation can be also a future direction. Moreover, a serious biophysical interpretation of the extracted motifs by DiMotif for nuclear localization signals, biofilm formation, and integrins/integrin

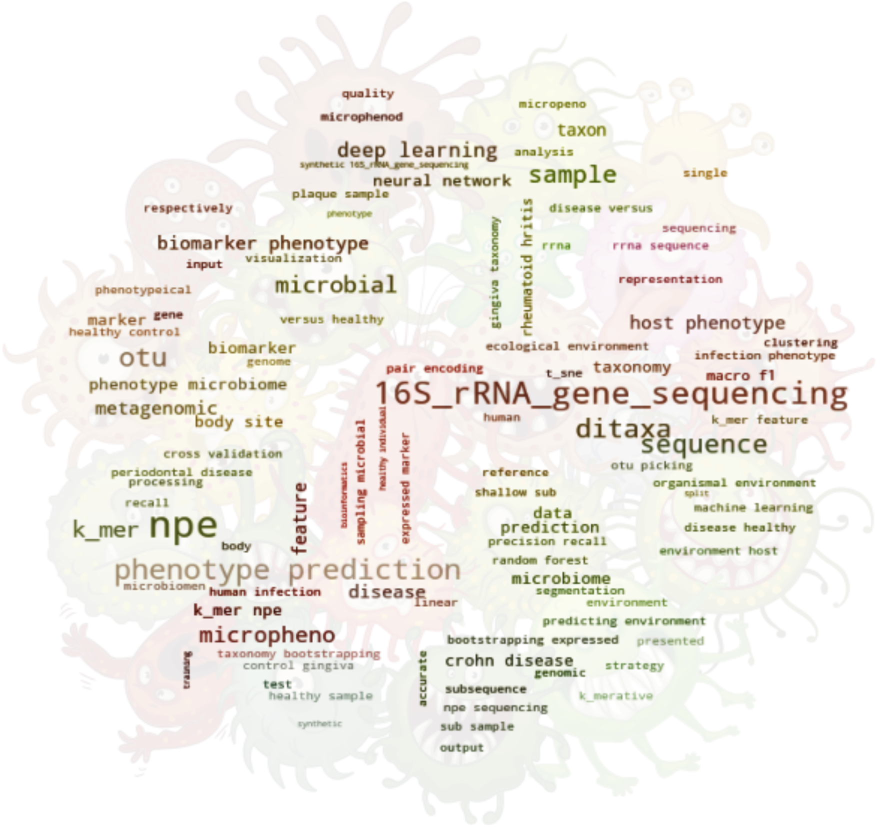


Figure 5.3. Semi-automatically generated word-cloud of the genomics/metagenomics-related contributions of the dissertation.
The background image is taken from https://www.welovesolo.com/funny-bacteria-cartoon-styles-vector-01/.

binding proteins have still remained elusive for a future study.

Structure prediction: One potential future direction for structure prediction is the prediction of PDB 3D structures (Rose et al. 2016) from the primary sequences. Recently, there has been great progress in this direction (AlQuraishi 2019). The precision of secondary structure datasets needs to be improved to avoid the significant errors we observe at the label transition locations. We explored data augmentation of one-hot vector based on the protein secondary structure prediction, which was not successful. A future direction could be exploring possible data augmentation schemes for the Position-Specific-Scoring-Matrix (PSSM) features.

2. Summary of life language processing for genomics/metagenomics and the future directions

A word-cloud summary of proteomics-related contributions is shown in Figure 5.3. Accurately resolving taxon-host disease relationships is required to elucidate the underlying functional mechanisms of taxon-host interactions in microbiome-linked diseases and establishing diagnosis and therapy options in precision medicine. As a result, accurate detection of disease phenotype and disease-associated biomarkers are among prominent problems in microbial informatics. Although shotgun metagenomic sequencing provides a better resolution in taxonomic assignment, 16S rRNA amplicon data, due to its low cost is still the most used data type in microbiome studies to date (Pollock et al. 2018). After 16S rRNA sequencing, reads are typically clustered based on their sequence similarity to each other, and the resulting clusters are referred to as operational taxonomic units (OTUs) having several disadvantages in the taxonomic assignment (detailed in Section §3.3). An alternative solution is the analysis of individual 16S rRNA gene sequence (Callahan et al. 2016; Amir et al. 2017; Nearing et al. 2018), which is computationally challenging, as each 16S rRNA sample may contain 10,000s of sequences. Here, we proposed two OTU-free 16S processing computational methods, one designed for rapid and accurate phenotype prediction, called MicroPheno, and the other, DiTaxa, which is targeted accurate phenotype and biomarker detection to find the phenotype-specific taxa as well. We proposed a bootstrapping framework to investigate the sufficiency of shallow subsamples in phenotype and biomarker detection, which is important for reducing the computational overhead as well as the cost of sequencing technology, i.e., not to proceed with unnecessary deep sequencing, which is more expensive.

Before metagenomics setting, in the prediction of ‘infection-causing’ strains of Klebsiella pneumoniae we showed that the k-mer based approaches can provide us with powerful phenotype predictors outperforming expensive genomic features. Next, we extend the k-mer based approach for the metagenomics setting (in MicroPheno) and subsequently, we extended the fixed length k-mers in metagenomics to variable length subsequences (in DiTaxa). Another genomics application covered in this thesis is the task of exonic region prediction discussed in §2.2, where the bio-vectors helped in more accurate labeling of exonic regions.

MicroPheno is a reference- and alignment-free method for predicting environments and host phenotypes from 16S rRNA gene sequencing based on k-mer representations. Deep learning methods, as well as classical approaches, were explored for predicting environments and host phenotypes. K-mer distribution of shallow sub-samples outperformed Operational Taxonomic Unit (OTU) features in the tasks of body-site identification and Crohn’s disease prediction. In addition, k-mer features predicted representative 16S rRNA gene sequences of 18 ecological environments and 5 organismal environments with high macro-F1 scores of 0.88 and 0.87.

DiTaxa is an alignment- and reference- free subsequence based 16S rRNA data processing, as a new paradigm for microbiome phenotype and biomarker detection. DiTaxa substituted standard OTU-clustering by segmenting 16S rRNA reads into the most frequent variable-length subsequences called nucleotide-pair encoding inspired from BPE, a data compression algorithm. We compared the performance of DiTaxa to the state-of-the-art methods in phenotype and biomarker detection, using human-associated 16S rRNA samples for periodontal disease, rheumatoid arthritis, and inflammatory bowel diseases, as well as a synthetic benchmark dataset. DiTaxa improved biomarker detection from microbial 16S rRNA datasets, in terms of accuracy, taxonomic resolution and computational efficiency over known links from literature and synthetic benchmark datasets, while performing competitively to the state-of-the-art

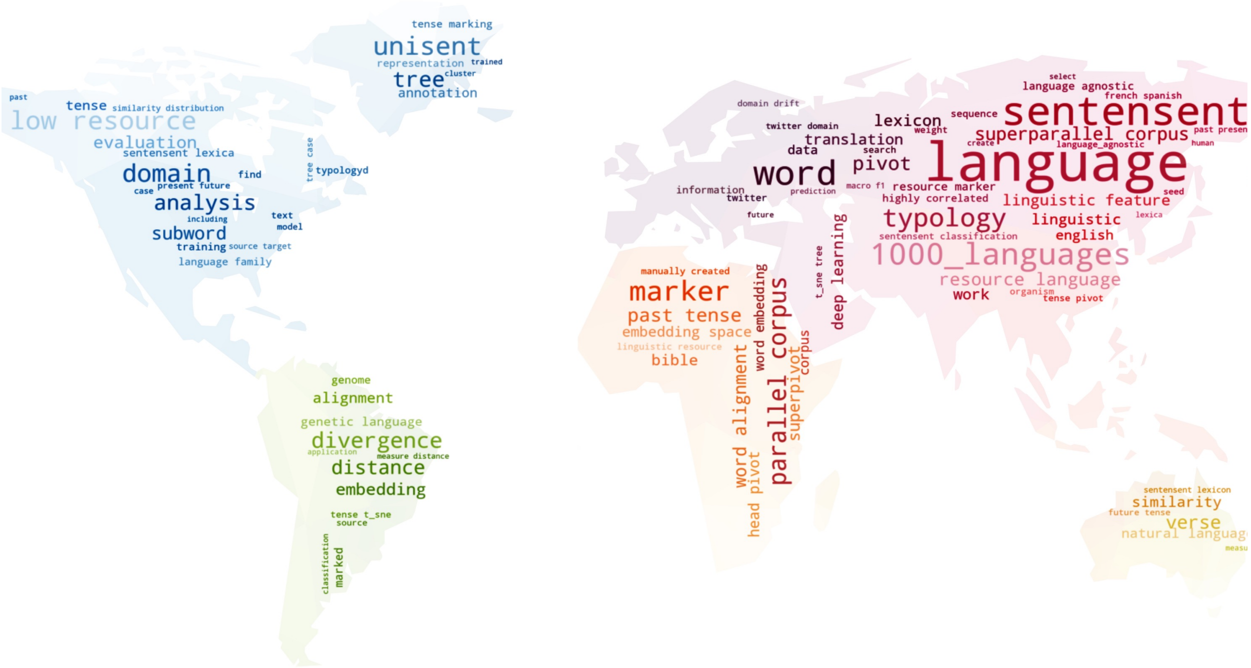


Figure 5.4. Semi-automatically generated word-cloud of the contributions in human language processing.
The background image is taken from
https://pasarelapr.com/images/language-world-map/language-world-map-2.png.

approach in phenotype prediction.

and als the genomics application - AMR prediction using NPE representation.

Future directions of life language processing in genomics/metagenomics

Shotgun metagenome analysis: Principally, applications of the NPE representation are not limited to 16S rRNA data but could include biomarker-discovery in genomics (e.g., antimicrobial resistance analysis), or from shotgun metagenome data, or other biological sequences, instead of using parameter-dependent clustering representations such as gene family assignments. Thus, the insights and methodology of DiTaxa may be expanded for genomics and shotgun metagenome data as well.

Cancer genomics: Cancer bioinformatics is another route for the expansion of life language processing project. The Cancer Genome Atlas (TCGA) (Tomczak et al. 2015) is an excellent data resource for this research containing petabytes of genomic and proteomic data to-date spanning more than thirty cancer types. The molecular characteristics-based diagnosis and treatment prediction are among problem settings that can be investigated within the life language processing framework.

3. Summary of life language processing for human languages and the future directions

A word-cloud summary of contributions in human language processing is shown in Figure 5.4. I introduced SuperPivot and word embedding based language divergence (WELD) for language-agnostic processing of natural languages, for linguistic marker detection and quantitative comparison of languages respectively. To the best of our knowledge, SuperPivot was the first large cross-lingual computational study performed on 1000 languages. SuperPivot only requires that a linguistic feature is overtly marked in a *few* of thousands of languages as opposed to requiring that it be marked in *all* languages under investigation. We showed that SuperPivot performed well on typological analysis of tense in 1000 languages. Subsequently, we used *SuperPivot* for the creation of *UniSent*, the largest sentiment resource to date in terms of the number of covered languages, including many low resource ones. Using *DomDrift*, a method to quantify domain drift for words in the UniSent given an embedding space in the target domain, we ensure domain adaptability of the *UniSent*. Next, we the domain shift measure in *DomDrift* and extend it for the whole embedding graph. We introduced word embedding language divergence, as a new method to quantify the high-level similarity of languages using their embedding graphs.

Future directions of life language processing for human languages

SuperPivot: SuperPivot at least has several potential future directions: (i) expanding the work for other linguistic markers; (ii) Addressing shortcomings of the way we compute alignments by developing a new method specific to the setting of bible parallel corpus, which is a very specific setting (many parallel languages, but languages are characteristically low-resource) in comparison with a usual scenario that thousands of parallel sentences exist only for a few languages.

Comparison of language/language-variations: The insight of word embedding language divergence may be expanded as an alternative approach for phylogenetic clustering of organisms. In addition, this method can be further developed for comparison and analysis of languages in the bible parallel corpora of 1000+ languages.

Bibliography

[1] Jørn A Aas et al. “Defining the normal bacterial flora of the oral cavity”. In: *Journal of clinical microbiology* 43.11 (2005), pp. 5721–5732.

[2] Amine Abdaoui et al. “Feel: French extended emotional lexicon”. In: *ELRA Catalogue of Language Resources. ISLRN* (2014), pp. 041–639.

[3] Loreto Abusleme et al. “The subgingival microbiome in health and periodontitis and its relationship with community biomass and inflammation”. In: *The ISME journal* 7.5 (2013), p. 1016.

[4] Heike Adel, Ehsaneddin Asgari, and Hinrich Schütze. “Overview of Character-Based Models for Natural Language Processing”. In: *International Conference on Computational Linguistics and Intelligent Text Processing*. Springer. 2017, pp. 3–16.

[5] Haithem Afli, Sorcha McGuire, and Andy Way. “Sentiment translation for low resourced languages: Experiments on irish general election tweets”. In: *18th International Conference on Computational Linguistics and Intelligent Text Processing*. 2017.

[6] Zeljko Agic et al. “Multilingual Projection for Parsing Truly Low-Resource Languages”. In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 301–312. URL: http://aclweb.org/anthology/Q16-1022.

[7] Judith L. Aissen. *Tzotzil Clause Structure*. Springer, 1987.

[8] Beatrice Alex. “An Unsupervised System for Identifying English Inclusions in German Text”. In: *Annual Meeting of the Association for Computational Linguistics*. 2005.

[9] Babak Alipanahi et al. “Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning”. In: *Nat. Biotechnol.* 33.8 (2015), pp. 831–838.

[10] Mohammed AlQuraishi. “AlphaFold at CASP13”. In: *Bioinformatics* (May 2019). ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz422. eprint: http://oup.prod.sis.lan/bioinformatics/advance-article-pdf/doi/10.1093/bioinformatics/btz422/28835737/btz422.pdf. URL: https://doi.org/10.1093/bioinformatics/btz422.

[11] Amnon Amir et al. “Deblur rapidly resolves single-nucleotide community sequence patterns”. In: *MSystems* 2.2 (2017), e00191–16.

[12] Roger W Andersen. “Papiamentu tense-aspect, with special attention to discourse”. In: *Pidgin and creole tense-mood-aspect systems* (1990), pp. 59–96.

[13] David Ando et al. “Physical motif clustering within intrinsically disordered nucleoporin sequences reveals universal functional features”. In: *PloS one* 8.9 (2013), e73831.

[14] Daniel Andor et al. “Globally Normalized Transition-Based Neural Networks”. In: *Annual Meeting of the Association for Computational Linguistics*. 2016.

[15] Christof Angermueller et al. “Deep learning for computational biology”. In: *Molecular systems biology* 12.7 (2016), p. 878.

[16] Florent E Angly et al. “Grinder: a versatile amplicon and shotgun sequence simulator”. In: *Nucleic acids research* 40.12 (2012), e94–e94.

[17] Ramon Aragues et al. “Characterization of protein hubs by inferring interacting motifs from protein interactions”. In: *PloS Computational Biology* 3.9 (2007), e178.

[18] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. “A simple but tough-to-beat baseline for sentence embeddings”. In: (2016).

[19] Marie-Claire Arrieta et al. “Early infancy microbial and metabolic alterations affect risk of childhood asthma”. In: *Science Translational Medicine* 7.307 (2015). ISSN: 1946-6242. DOI: 10.1126/scitranslmed.aab2271. URL: http://stm.sciencemag.org/scitransmed/7/307/307ra152.full.pdf.

[20] Ehsaneddin Asgari, Fabienne Braune, et al. “UniSent: Universal Adaptable Sentiment Lexica for 1000+ Languages”. In: *arXiv preprint arXiv:1904.09678* (2019).

[21] Ehsaneddin Asgari and Jean-Cédric Chappelier. “Linguistic resources and topic models for the analysis of persian poems”. In: *Proceedings of the Workshop on Computational Linguistics for Literature*. 2013, pp. 23–31.

[22] Ehsaneddin Asgari, Kiavash Garakani, et al. “MicroPheno: predicting environments and host phenotypes from 16S rRNA gene sequencing using a k-mer based representation of shallow sub-samples”. In: *Bioinformatics* 34.13 (2018), pp. i32–i42. DOI: 10.1093/bioinformatics/bty296.

[23] Ehsaneddin Asgari, Marzyeh Ghassemi, and Mark Alan Finlayson. “Confirming the themes and interpretive unity of Ghazal poetry using topic models”. In: *Neural Information Processing Systems (NIPS) Workshop for Topic Models*. 2013.

[24] Ehsaneddin Asgari and Mohammad R. K. Mofrad. “Comparing Fifty Natural Languages and Twelve Genetic Languages Using Word Embedding Language Divergence (WELD) as a Quantitative Measure of Language Distance”. In: *In Proceedings of the NAACL-HLT Workshop on Multilingual and Cross-lingual Methods in NLP, San Diego, CA*. Association for Computational Linguistics. 2016, pp. 65–74.

[25] Ehsaneddin Asgari and Mohammad RK Mofrad. “Continuous distributed representation of biological sequences for deep proteomics and genomics”. In: *PloS One* 10.11 (2015), e0141287.

[26] Ehsaneddin Asgari, Philipp C Münch, et al. “DiTaxa: nucleotide-pair encoding of 16S rRNA for host phenotype and biomarker detection”. In: *Bioinformatics* (Nov. 2018). ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty954. eprint: http://oup.prod.sis.lan/bioinformatics/advance-article-pdf/doi/10.1093/bioinformatics/bty954/27452903/bty954.pdf. URL: https://doi.org/10.1093/bioinformatics/bty954.

[27] Ehsaneddin Asgari, Soroush Nasiriany, and Mohammad RK Mofrad. “Text Analysis and Automatic Triage of Posts in a Mental Health Forum”. In: *Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology*. 2016, pp. 153–157.

[28] Ehsaneddin Asgari and Ali Sanaei. “Measuring Countries’ Human Rights Positions in UN Universal Periodic Review”. In: *Available at SSRN* (2017). URL: http://dx.doi.org/10.2139/ssrn.3029031.

[29] Ehsaneddin Asgari and Hinrich Schütze. “Past, Present, Future: A Computational Investigation of the Typology of Tense in 1000 Languages”. In: *EMNLP*. 2017.

[30] Erki Aun et al. “A k-mer-based method for the identification of phenotype-associated genomic biomarkers and predicting phenotypes of sequenced bacteria”. In: *PLoS computational biology* 14.10 (2018), e1006434.

[31] Akinori Awazu. “Prediction of nucleosome positioning by the incorporation of frequencies and distributions of three different nucleotide segment lengths into a general pseudo k-tuple nucleotide composition”. In: *Bioinformatics* 33.1 (2016), pp. 42–48.

[32] Mohammad Azimi and Mohammad RK Mofrad. “Higher Nucleoporin-Importinβ Affinity at the Nuclear Basket Increases Nucleocytoplasmic Import”. In: *PloS one* 8.11 (2013), e81741.

[33] Dzmitry Bahdanau et al. “End-to-end attention-based large vocabulary speech recognition”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2016, pp. 4945–4949.

[34] Timothy L Bailey et al. “MEME SUITE: Tools for motif discovery and searching”. In: *Nucleic Acids Res.* 37.suppl_2 (2009), W202–W208.

[35] Timothy Baldwin and Marco Lui. “Language Identification: The Long and the Short of the Matter”. In: *Conference of the North American Chapter of the Association for Computational Linguistics / Human Language Technologies*. 2010, pp. 229–237.

[36] Miguel Ballesteros, Chris Dyer, and Noah A. Smith. “Improved transition-based parsing by modeling characters instead of words with LSTMs”. In: *Conference on Empirical Methods in Natural Language Processing*. 2015.

[37] Anton Bankevich et al. “SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing”. In: *Journal of computational biology* 19.5 (2012), pp. 455–477.

[38] Colin Bannard and Chris Callison-Burch. “Paraphrasing with bilingual parallel corpora”. In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 2005, pp. 597–604.

[39] Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. “Bilingual Sentiment Embeddings: Joint Projection of Sentiment Across Languages”. In: *ACL*. 2018.

[40] Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. “Projecting Embeddings for Domain Adaptation: Joint Modeling of Sentiment Analysis in Diverse Domains”. In: *COLING*. 2018.

[41] Emily M Bender. “Linguistically naive!= language independent: why NLP needs linguistic typology”. In: *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?* Association for Computational Linguistics. 2009, pp. 26–32.

[42] Emily M Bender. “On achieving and evaluating language-independence in NLP”. In: *Linguistic Issues in Language Technology* 6.3 (2011), pp. 1–26.

[43] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828. ISSN: 01628828. DOI: 10.1109/TPAMI.2013.50. arXiv: 1206.5538.

[44] Taylor Berg-Kirkpatrick and Dan Klein. “Phylogenetic grammar induction”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2010, pp. 1288–1297.

[45] Axel Bernal et al. “Global discriminative learning for higher-accuracy computational gene prediction”. In: *PLoS Comput Biol* 3.3 (2007), e54.

[46] Michael Bernhofer et al. “NLSdb—Major update for database of nuclear localization signals and nuclear export signals”. In: *Nucleic Acids Res.* 46.D1 (2017), pp. D503–D508.

[47] Jeff Bilmes and Katrin Kirchhoff. “Factored Language Models and Generalized Parallel Backoff”. In: *Conference of the North American Chapter of the Association for Computational Linguistics / Human Language Technologies*. 2003.

[48] Laure B Bindels et al. “Synbiotic approach restores intestinal homeostasis and prolongs survival in leukaemic mice with cachexia”. In: *The ISME journal* 10.6 (2016), p. 1456.

[49] Steven Bird. “NLTK: the natural language toolkit”. In: *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics. 2006, pp. 69–72.

[50] Maximilian Bisani and Hermann Ney. “Joint-sequence models for grapheme-to-phoneme conversion”. In: *Speech Communication* 50.5 (2008), pp. 434–451.

[51] David M Blei. “Probabilistic topic models”. In: *Communications of the ACM* 55.4 (2012), pp. 77–84.

[52] Bernd Bohnet and Joakim Nivre. "A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics. 2012, pp. 1455–1465.

[53] Bernd Bohnet, Joakim Nivre, et al. "Joint morphological and syntactic analysis for richly inflected languages". In: *Transactions of the Association for Computational Linguistics* 1 (2013), pp. 415–428.

[54] Piotr Bojanowski et al. "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* (2017).

[55] Johan Bollen, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market". In: *Journal of computational science* 2.1 (2011), pp. 1–8.

[56] Peer Bork et al. "Predicting function: from genes to genomes and back". In: *Journal of molecular biology* 283.4 (1998), pp. 707–725.

[57] Jan A Botha and Phil Blunsom. "Compositional Morphology for Word Representations and Language Modelling". In: *International Conference on Machine Learning*. 2014.

[58] Alexandre Bouchard-Cote et al. "Automated reconstruction of ancient languages using probabilistic models of sound change". In: *Proceedings of the National Academy of Sciences* 110.11 (2013), pp. 4224–4229.

[59] Emmanuel Boutet et al. "UniProtKB/Swiss-Prot, the manually annotated section of the UniProt KnowledgeBase: How to use the entry view". In: *Plant Bioinformatics*. Springer, 2016, pp. 23–54.

[60] L. Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 08856125.

[61] Peter F Brown et al. "The mathematics of statistical machine translation: Parameter estimation". In: *Computational linguistics* 19.2 (1993), pp. 263–311.

[62] Joan L Bybee and Oesten Dahl. *The creation of tense and aspect systems in the languages of the world*. John Benjamins Amsterdam, 1989.

[63] CZ Cai et al. "SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence". In: *Nucleic acids research* 31.13 (2003), pp. 3692–3697.

[64] Yunpeng Cai et al. "ESPRIT-Forest: Parallel clustering of massive amplicon sequence data in subquadratic time". In: *PLoS Computational Biology* 13.4 (2017). ISSN: 15537358. DOI: 10.1371/journal.pcbi.1005518.

[65] Benjamin J Callahan et al. "DADA2: high-resolution sample inference from Illumina amplicon data". In: *Nature methods* 13.7 (2016), p. 581.

[66] Christiam Camacho et al. "BLAST+: architecture and applications". In: *BMC bioinformatics* 10.1 (2009), p. 421.

[67] Giovanni Cammarota, Gianluca Ianiro, and Antonio Gasbarrini. *Fecal microbiota transplantation for the treatment of clostridium difficile infection: A systematic review.* 2014. DOI: 10.1097/MCG.0000000000000046.

[68] Ramon Ferrer i Cancho and Richard V Sole. "The small world of human language". In: *Proceedings of the Royal Society of London B: Biological Sciences* 268.1482 (2001), pp. 2261–2265.

[69] Kris Cao and Marek Rei. "A Joint Model for Word Embedding and Word Morphology". In: *Annual Meeting of the Association for Computational Linguistics*. 2016, pp. 18–26.

[70] J. Gregory Caporaso et al. *QIIME allows analysis of high-throughput community sequencing data*. 2010. DOI: 10.1038/nmeth.f.303. arXiv: NIHMS150003.

[71] Anna Paola Carrieri, Niina Haiminen, and Laxmi Parida. "Host Phenotype Prediction from Differentially Abundant Microbes Using RoDEO". In: *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*. Springer. 2016, pp. 27–41. DOI: 10.1007/978-3-319-67834-4_3.

[72] George Casella and Roger L. Berger. *Statistical Inference*. Thomson, 2008.

[73] W Cavnar. "Using an n-gram-based document representation with a vector processing retrieval model". In: *NIST SPECIAL PUBLICATION SP* (1995), pp. 269–269.

[74] William Chan et al. "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2016, pp. 4960–4964.

[75] Jacqueline M. Chaparro et al. *Manipulating the soil microbiome to increase soil health and plant fertility*. 2012. DOI: 10.1007/s00374-012-0691-4.

[76] Aitao Chen et al. "Chinese text retrieval without using a dictionary". In: *ACM SIGIR Forum* 31.SI (1997), pp. 42–49.

[77] Lei Chen, Shiyong Lu, and Jeffrey Ram. "Compressed pattern matching in DNA sequences". In: *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*. IEEE. 2004, pp. 62–68.

[78] Xinxiong Chen et al. "Joint Learning of Character and Word Embeddings". In: *International Joint Conference on Artificial Intelligence*. 2015, pp. 1236–1242.

[79] Yanqing Chen and Steven Skiena. "Building Sentiment Lexicons for All Major Languages". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 383–389. DOI: 10.3115/v1/P14-2063. URL: https://www.aclweb.org/anthology/P14-2063.

[80] Jason P. C. Chiu and Eric Nichols. "Named Entity Recognition with Bidirectional LSTM-CNNs". In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 357–370.

[81] Ilseung Cho and Martin J. Blaser. *The human microbiome: At the interface of health and disease*. 2012. DOI: 10.1038/nrg3182. arXiv: NIHMS150003.

[82] Noam Chomsky. *Syntactic structures*. Walter de Gruyter, 2002.

[83] Peter Y Chou and Gerald D Fasman. "Prediction of protein conformation". In: *Biochemistry* 13.2 (1974), pp. 222–245.

[84] Gobinda G Chowdhury. "Natural language processing". In: *Annual review of information science and technology* 37.1 (2003), pp. 51–89.

[85] Christos Christodouloupoulos and Mark Steedman. "A massively parallel corpus: the bible in 100 languages". In: *Language resources and evaluation* 49.2 (2015), pp. 375–395.

[86] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. "Hierarchical Multiscale Recurrent Neural Networks". In: *Proceedings of International Conference on Learning Representations*. 2017.

[87] Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. "A Character-level Decoder without Explicit Segmentation for Neural Machine Translation". In: *Annual Meeting of the Association for Computational Linguistics*. 2016.

[88] Junyoung Chung, Caglar Gulcehre, et al. "Gated feedback recurrent neural networks". In: *International Conference on Machine Learning*. 2015, pp. 2067–2075.

[89] Kenneth Ward Church. "Char_align: A Program for Aligning Parallel Texts at the Character Level". In: *Annual Meeting of the Association for Computational Linguistics*. 1993, pp. 1–8.

[90] Mark Cieliebak et al. "A Twitter corpus and benchmark resources for German sentiment analysis". In: *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. 2017, pp. 45–51.

[91] Christopher Cieri et al. "Selection Criteria for Low Resource Language Programs." In: *LREC*. 2016.

[92] Alina Maria Ciobanu and Liviu P. Dinu. "An Etymological Approach to Cross-Language Orthographic Similarity. Application on Romanian". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1047–1058. URL: http://www.aclweb.org/anthology/D14-1112.

[93] Alexander Clark. "Combining distributional and morphological information for part of speech induction". In: *Conference of the European Chapter of the Association for Computational Linguistics*. 2003, pp. 59–66.

[94] Wyatt T Clark and Predrag Radivojac. "Analysis of protein function and its prediction from amino acid sequence". In: *Proteins: Structure, Function, and Bioinformatics* 79.7 (2011), pp. 2086–2096.

[95] Ronan Collobert and Jason Weston. “A unified architecture for natural language processing: Deep neural networks with multitask learning”. In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 160–167.

[96] Ronan Collobert, Jason Weston, et al. “Natural Language Processing (Almost) from Scratch”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2493–2537.

[97] Bernard Comrie. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press, 1989.

[98] Jin Cong and Haitao Liu. “Approaching human language with complex networks”. In: *Physics of life reviews* 11.4 (2014), pp. 598–618.

[99] Alexis Conneau et al. “Word translation without parallel data”. In: *arXiv preprint arXiv:1710.04087* (2017).

[100] Gene Ontology Consortium. “The Gene Ontology resource: 20 years and still GOing strong”. In: *Nucleic acids research* 47.D1 (2018), pp. D330–D338.

[101] UniProt Consortium. “UniProt: the universal protein knowledgebase”. In: *Nucleic Acids Res.* 45.D1 (2016), pp. D158–D169.

[102] Geoffrey M Cooper, Robert E Hausman, and Robert E Hausman. *The cell: a molecular approach*. Vol. 10. ASM press Washington, DC, 2000.

[103] Tristan Cordier et al. “Predicting the Ecological Quality Status of Marine Environments from eDNA Metabarcoding Data Using Supervised Machine Learning”. In: *Environmental Science & Technology* 51.16 (2017). ISSN: 0013-936X. DOI: 10.1021/acs.est.7b01518. URL: http://pubs.acs.org/doi/abs/10.1021/acs.est.7b01518.

[104] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine learning* 20.3 (1995), pp. 273–297.

[105] Luciano da Fontoura Costa et al. “Analyzing and modeling real-world phenomena with complex networks: a survey of applications”. In: *Advances in Physics* 60.3 (2011), pp. 329–412.

[106] Marta R. Costa-Jussa and Jose A. R. Fonollosa. “Character-based Neural Machine Translation”. In: *Annual Meeting of the Association for Computational Linguistics*. 2016.

[107] Elizabeth K Costello et al. “Bacterial community variation in human body habitats across space and time.” In: *Science (New York, N.Y.)* 326.5960 (2009), pp. 1694–7. DOI: 10.1126/science.1177486.

[108] Ryan Cotterell, Tim Vieira, and Hinrich Schuetze. “A Joint Model of Orthography and Morphological Segmentation”. In: *Conference of the North American Chapter of the Association for Computational Linguistics / Human Language Technologies*. 2016.

[109] Fabien Cottier et al. “Advantages of meta-total RNA sequencing (MeTRS) over shotgun metagenomics and amplicon-based sequencing in the profiling of complex microbial communities”. In: *npj Biofilms and Microbiomes* 4.1 (2018), p. 2.

[110] JS Cramer. *The Origins of Logistic Regression: Tinbergen Institute Discussion Papers*. Tech. rep. No 02-119/4, Tinbergen Institute, 2002.

[111] William Croft. *Radical construction grammar: Syntactic theory in typological perspective*. Oxford University Press on Demand, 2001.

[112] William Croft. *Typology and universals*. Cambridge University Press, 2002.

[113] William Croft and Keith T Poole. "Inferring universals from grammatical variation: Multidimensional scaling for typological analysis". In: *Theoretical linguistics* 34.1 (2008), pp. 1–37.

[114] Hongfei Cui and Xuegong Zhang. "Alignment-free supervised classification of metagenomes by recursive SVM". In: *BMC Genomics* 14.1 (2013). ISSN: 14712164. DOI: 10.1186/1471-2164-14-641.

[115] Adele Cutler, D Richard Cutler, and John R Stevens. "Random forests". In: *Ensemble Machine Learning*. Springer, 2012, pp. 157–175.

[116] Michael Cysouw. "Inducing semantic roles". In: *Perspectives on semantic roles* (2014), pp. 23–68.

[117] Michael Cysouw and Bernhard Waelchli. "Parallel texts: using translational equivalents in linguistic typology". In: *STUF-Sprachtypologie und Universalienforschung* 60.2 (2007), pp. 95–99.

[118] Oesten Dahl. "From questionnaires to parallel corpora in typology". In: *STUF-Sprachtypologie und Universalienforschung* 60.2 (2007), pp. 172–181.

[119] Oesten Dahl. *Tense and Aspect in the Languages of Europe*. Walter de Gruyter, 2000.

[120] Oesten Dahl. *Tense and aspect systems*. Basil Blackwell, 1985.

[121] Oesten Dahl. "The perfect map: Investigating the cross-linguistic distribution of TAME categories in a parallel corpus". In: *Aggregating Dialectology, Typology, and Register Contents Analysis. Linguistic Variation in Text and Speech. Linguae & litterae* 28 (2014), pp. 268–289.

[122] M. Damashek. "Gauging similarity with N-grams. Language-independent categorization of text". In: *Science* 267 (1995), pp. 843–848.

[123] Maria Carlota Dao et al. "Akkermansia muciniphila and improved metabolic health during a dietary intervention in obesity: Relationship with gut microbiome richness and ecology". In: *Gut* 65.3 (2016), pp. 426–436. ISSN: 14683288. DOI: 10.1136/gutjnl-2014-308778.

[124] Mohammad Darwich, Shahrul Azman Mohd Noah, and Nazlia Omar. "Minimally-supervised sentiment lexicon induction model: A case study of malay sentiment analysis". In: *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*. Springer. 2017, pp. 225–237.

[125] Dipanjan Das and Slav Petrov. "Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 600–609. URL: http://www.aclweb.org/anthology/P11-1061.

[126] Harold Charles Daume. *Practical structured learning techniques for natural language processing*. ProQuest, 2006.

[127] Norman E Davey et al. "SLiMSearch 2.0: biological context for short linear motifs in proteins". In: *Nucleic Acids Res.* 39.suppl_2 (2011), W56–W60.

[128] T De Heer. "Experiments with syntactic traces in information retrieval". In: *Information Storage and Retrieval* 10.3-4 (1974), pp. 133–144.

[129] Xin Deng and Jianlin Cheng. "MSACompro: protein multiple sequence alignment using predicted secondary structure, solvent accessibility, and residue-residue contacts". In: *BMC bioinformatics* 12.1 (2011), p. 472.

[130] Zhi Luo Deng et al. "Dysbiosis in chronic periodontitis: Key microbial players and interactions with the human host". In: *Scientific Reports* 7.1 (2017), pp. 1–13. ISSN: 20452322. DOI: 10.1038/s41598-017-03804-8.

[131] Jan Deriu et al. "Leveraging large amounts of weakly supervised data for multi-language sentiment classification". In: *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee. 2017, pp. 1045–1052.

[132] T. Z. DeSantis et al. "Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB". In: *Applied and Environmental Microbiology* 72.7 (2006), pp. 5069–5072. ISSN: 00992240. DOI: 10.1128/AEM.03006-05.

[133] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[134] Mona Diab and Philip Resnik. "An Unsupervised Method for Word Sense Tagging using Parallel Corpora". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. 2002. URL: http://www.aclweb.org/anthology/P02-1033.

[135] Holger Dinkel et al. "ELM—the database of eukaryotic linear motifs". In: *Nucleic Acids Res.* 40.D1 (2011), pp. D242–D251.

[136] Gregory Ditzler, Robi Polikar, and Gail Rosen. "Multi-Layer and Recursive Neural Networks for Metagenomic Classification". In: *IEEE Transactions on Nanobioscience* 14.6 (2015). ISSN: 15361241. DOI: 10.1109/TNB.2015.2461219.

[137] Shan Dong and David B Searls. "Gene structure prediction by linguistic methods". In: *Genomics* 23.3 (1994), pp. 540–551.

[138] Matthew S Dryer et al. *The world atlas of language structures.* Oxford University Press, 2005.

[139] Jingcheng Du et al. "Gene2Vec: Distributed Representation of Genes Based on Co-Expression". In: *bioRxiv* (2018), p. 286096.

[140] A Keith Dunker et al. "Function and structure of inherently disordered proteins". In: *Current opinion in structural biology* 18.6 (2008), pp. 756–764.

[141] Ted Dunning. *Statistical identification of language.* Tech. rep. MCCS 940-273. Computing Research Laboratory, New Mexico State, 1994.

[142] Bas E. Dutilh et al. "Explaining microbial phenotypes on a genomic scale: Gwas for microbes". In: *Briefings in Functional Genomics* 12.4 (2013), pp. 366–0380. ISSN: 20412649. DOI: 10.1093/bfgp/elt008.

[143] Aparajita Dutta et al. "SpliceVec: distributed feature representations for splice junction prediction". In: *Computational biology and chemistry* 74 (2018), pp. 434–441.

[144] Claire Duvallet et al. "Meta-analysis of gut microbiome studies identifies disease-specific and shared responses". In: *Nature Communications* 8.1 (2017), p. 1784. ISSN: 2041-1723. DOI: 10.1038/s41467-017-01973-8. URL: http://www.nature.com/articles/s41467-017-01973-8.

[145] Chris Dyer, Victor Chahuneau, and Noah A. Smith. "A Simple, Fast, and Effective Reparameterization of IBM Model 2". In: *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA.* 2013, pp. 644–648.

[146] H Jane Dyson and Peter E Wright. "Intrinsically unstructured proteins and their functions". In: *Nature reviews Molecular cell biology* 6.3 (2005), pp. 197–208.

[147] A. Eck et al. "Robust microbiota-based diagnostics for inflammatory bowel disease". In: *Journal of Clinical Microbiology* 55.6 (2017), pp. 1720–1732. ISSN: 1098660X. DOI: 10.1128/JCM.00162-17.

[148] Robert C Edgar. "UPARSE: highly accurate OTU sequences from microbial amplicon reads". In: *Nature methods* 10.10 (2013), p. 996.

[149] Robert C. Edgar et al. "UCHIME improves sensitivity and speed of chimera detection". In: *Bioinformatics* 27.16 (2011), pp. 2194–2200. ISSN: 13674803. DOI: 10.1093/bioinformatics/btr381.

[150] Richard J Edwards, Norman E Davey, and Denis C Shields. "SLiMFinder: a probabilistic method for identifying over-represented, convergently evolved, short linear motifs in proteins". In: *PloS one* 2.10 (2007), e967.

[151] Olof Emanuelsson et al. "Locating proteins in the cell using TargetP, SignalP and related tools". In: *Nat. Protoc.* 2.4 (2007), pp. 953–971.

[152] Guy Emerson et al. "Seedling: Building and using a seed corpus for the human language project". In: *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*. 2014, pp. 77–85.

[153] Emilio A Emini et al. "Induction of hepatitis A virus-neutralizing antibody by a virus-specific synthetic peptide." In: *J. Virology* 55.3 (1985), pp. 836–839.

[154] Anton J Enright, Stijn Van Dongen, and Christos A Ouzounis. "An efficient algorithm for large-scale detection of protein families". In: *Nucleic acids research* 30.7 (2002), pp. 1575–1584.

[155] Florian Eyben et al. "From speech to letters - using a novel neural network architecture for grapheme based ASR". In: *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*. 2009, pp. 376–380.

[156] Asli Eyecioglu and Bill Keller. "ASOBEK at SemEval-2016 Task 1: Sentence representation with character N-gram embeddings for semantic textual similarity". In: *SemEval-2016: The 10th International Workshop on Semantic Evaluation*. 2016, pp. 1320–1324.

[157] Manaal Faruqui et al. "Morphological Inflection Generation Using Character Sequence to Sequence Learning". In: *Conference of the North American Chapter of the Association for Computational Linguistics / Human Language Technologies*. 2016.

[158] Noah Fierer. *Embracing the unknown: Disentangling the complexities of the soil microbiome*. 2017. DOI: 10.1038/nrmicro.2017.87.

[159] Noah Fierer et al. "Forensic identification using skin bacterial communities." In: *Proceedings of the National Academy of Sciences of the United States of America* 107.14 (2010), pp. 6477–81. ISSN: 1091-6490. DOI: 10.1073/pnas.1000162107. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2852011%7B%5C&%7Dtool=pmcentrez%7B%5C&%7Drendertype=abstract%7B%5C%%7D5Cnhttp://www.pnas.org/content/107/14/6477.short.

[160] AV Finkelstein and OB Ptitsyn. "Statistical analysis of the correlation among amino acid residues in helical, β-structural and non-regular regions of globular proteins". In: *Journal of molecular biology* 62.3 (1971), pp. 613–624.

[161] Robert D Finn et al. "Pfam: the protein families database". In: *Nucleic acids research* (2013), gkt1223.

[162] John Rupert Firth. *Modes of meaning*. College Division of Bobbs-Merrill Company, 1975.

[163] Lukas Folkman et al. "DDIG-in: detecting disease-causing genetic variations due to frameshifting indels and nonsense mutations employing sequence and structural properties at nucleotide and protein levels". In: *Bioinformatics* 31.10 (2015), pp. 1599–1606.

[164] Alexander Fraser and Daniel Marcu. “Measuring word alignment quality for statistical machine translation”. In: *Computational Linguistics* 33.3 (2007), pp. 293–303.

[165] Annemarie Friedrich and Damyana Gateva. “Classification of telicity using cross-linguistic annotation projection”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 2559–2565. URL: http://aclweb.org/anthology/D17-1271.

[166] Martin C Frith et al. “Discovering sequence motifs with arbitrary insertions and deletions”. In: *PLoS Compu. Biol.* 4.5 (2008), e1000071.

[167] Cesar de la Fuente-Nuñez and Timothy K. Lu. “CRISPR-Cas9 technology: applications in genome engineering, development of sequence-specific antimicrobials, and future prospects”. In: *Integr. Biol.* 9.2 (2017), pp. 109–122. ISSN: 1757-9694. DOI: 10.1039/C6IB00140H. URL: http://xlink.rsc.org/?DOI=C6IB00140H.

[168] Kunihiko Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological cybernetics* 36.4 (1980), pp. 193–202.

[169] Ranko Gacesa, David J Barlow, and Paul F Long. “Machine learning can differentiate venom toxins from other proteins having non-toxic physiological functions”. In: *PeerJ Comput. Sci.* 2 (2016), e90.

[170] Philip Gage. “A new algorithm for data compression”. In: *The C Users Journal* 12.2 (1994), pp. 23–38.

[171] Madhavi Ganapathiraju et al. “Comparative n-gram analysis of whole-genome protein sequences”. In: *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc. 2002, pp. 76–81.

[172] Yuyang Gao et al. “Comparison of directed and weighted co-occurrence networks of six languages”. In: *Physica A: Statistical Mechanics and its Applications* 393 (2014), pp. 579–589.

[173] Stefan Gerdjikov and Klaus U Schulz. “Corpus analysis without prior linguistic knowledge-unsupervised mining of phrases and subphrase structure”. In: *CoRR* abs/1602.05772 (2016).

[174] Dirk Gevers et al. “The treatment-naive microbiome in new-onset Crohn’s disease”. In: *Cell Host and Microbe* 15.3 (2014), pp. 382–392. ISSN: 19346069. DOI: 10.1016/j.chom.2014.02.005.

[175] Raffaele Giancarlo, Simona E Rombo, and Filippo Utro. “Epigenomic k-mer dictionaries: shedding light on how sequence composition influences in vivo nucleosome positioning”. In: *Bioinformatics* 31.18 (2015), pp. 2939–2946.

[176] Jack A. Gilbert and Josh D. Neufeld. “Life in a World without Microbes”. In: *PLoS Biology* 12.12 (2014). ISSN: 15457885. DOI: 10.1371/journal.pbio.1002020.

[177] Dan Gillick et al. "Multilingual Language Processing From Bytes". In: *North American Chapter of the Association for Computational Linguistics*. June 2016, pp. 1296–1306.

[178] Ciara Gimblet et al. "Cutaneous Leishmaniasis Induces a Transmissible Dysbiotic Skin Microbiota that Promotes Skin Inflammation". In: *Cell Host and Microbe* 22.1 (2017), 13–24.e4. ISSN: 19346069. DOI: 10.1016/j.chom.2017.06.006.

[179] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Domain adaptation for large-scale sentiment classification: A deep learning approach". In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, pp. 513–520.

[180] David Golub and Xiadong He. "Character-Level Question Answering with Attention". In: *Conference on Empirical Methods in Natural Language Processing*. 2016.

[181] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[182] Manfred G Grabherr et al. "Full-length transcriptome assembly from RNA-Seq data without a reference genome". In: *Nat. Biotechnol.* 29.7 (2011), pp. 644–652.

[183] Alex Graves. "Generating Sequences With Recurrent Neural Networks". In: *CoRR* abs/1308.0850 (2013).

[184] Alex Graves and Navdeep Jaitly. "Towards End-To-End Speech Recognition with Recurrent Neural Networks". In: *International Conference on Machine Learning*. 2014, pp. 1764–1772.

[185] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks". In: *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE. 2013, pp. 6645–6649.

[186] Joseph H Greenberg. "A quantitative approach to the morphological typology of language". In: *International journal of American linguistics* 26.3 (1960), pp. 178–194.

[187] Jun-Lin Guan and Richard O Hynes. "Lymphoid cells recognize an alternatively spliced segment of fibronectin via the integrin receptor $\alpha4\beta1$". In: *Cell* 60.1 (1990), pp. 53–61.

[188] Kunchur Guruprasad, BV Bhasker Reddy, and Madhusudan W Pandit. "Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence". In: *Protein Eng. Des. Sel.* 4.2 (1990), pp. 155–161.

[189] Li Haizhou, Zhang Min, and Su Jian. "A joint source-channel model for machine transliteration". In: *Annual Meeting of the Association for Computational Linguistics*. 2004, p. 159.

[190] Jan Hajic, Jan Hric, and Vladislav Kubon. "Machine translation of very close languages". In: *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics. 2000, pp. 7–12.

[191] George Hajishengallis et al. “Low-abundance biofilm species orchestrates inflammatory periodontal disease through the commensal microbiota and complement”. In: *Cell host & microbe* 10.5 (2011), pp. 497–506.

[192] David Hall and Dan Klein. “Finding cognate groups using phylogenies”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2010, pp. 1030–1039.

[193] Micah Hamady and Rob Knight. *Microbial community profiling for human microbiome projects: Tools, techniques, and challenges*. 2009. DOI: 10.1101/gr.085464.108. arXiv: 0402594v3 [arXiv:cond-mat].

[194] Md Nafiz Hamid and Iddo Friedberg. “Identifying Antimicrobial Peptides using Word Embedding with Deep Recurrent Neural Networks”. In: *bioRxiv* (2018), p. 255505.

[195] William L Hamilton et al. “Inducing domain-specific sentiment lexicons from unlabeled corpora”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*. Vol. 2016. NIH Public Access. 2016, p. 595.

[196] Lushan Han et al. “Improving word similarity by augmenting PMI with estimates of word polysemy”. In: *Knowledge and Data Engineering, IEEE Transactions on* 25.6 (2013), pp. 1307–1322.

[197] Viktor Hangya et al. “Two Methods for Domain Adaptation of Bilingual Tasks: Delightfully Simple and Broadly Applicable”. In: *ACL*. 2018.

[198] Christian Hardmeier. “A Neural Model for Part-of-Speech Tagging in Historical Texts”. In: *International Conference on Computational Linguistics*. 2016, pp. 922–931.

[199] Iren Hartmann, Martin Haspelmath, and Michael Cysouw. “Identifying semantic role clusters and alignment types via microrole coexpression tendencies”. In: *Studies in Language. International Journal sponsored by the Foundation “Foundations of Language”* 38.3 (2014), pp. 463–484.

[200] Bo He et al. “Predicting intrinsic disorder in proteins: an overview”. In: *Cell research* 19.8 (2009), pp. 929–949.

[201] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[202] Yan He et al. “Erratum to: Stability of operational taxonomic units: an important but neglected property for analyzing microbial diversity”. In: *Microbiome* 3.1 (2015). ISSN: 2049-2618. DOI: 10.1186/s40168-015-0098-1. URL: http://www.microbiomejournal.com/content/3/1/34.

[203] Serge Heiden et al. “Typtex: Inductive typological text classification by multivariate statistical analysis for nlp systems tuning/evaluation”. In: *Maria Gavrilidou, George Carayannis, Stella Markantonatou, Stelios Piperidis, Gregory Stainhaouer (eds) Second International Conference on Language Resources and Evaluation*. 2000, p–141.

[204] Falk Hildebrand et al. "LotuS: an efficient and user-friendly OTU processing pipeline". In: *Microbiome* 2.1 (2014), p. 30.

[205] Geoffrey E Hinton. "Distributed representations". In: *Computer Science Department, Carnegie Mellon University* (1984).

[206] Teemu Hirsimaki et al. "Unlimited vocabulary speech recognition with morph language models applied to Finnish". In: *Computer Speech & Language* 20.4 (2006), pp. 515–541.

[207] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[208] Alexander Hogenboom et al. "Exploiting emoticons in sentiment analysis". In: *Proceedings of the 28th annual ACM symposium on applied computing*. ACM. 2013, pp. 703–710.

[209] Daniel J Hopkins and Gary King. "A method of automated nonparametric content analysis for social science". In: *American Journal of Political Science* 54.1 (2010), pp. 229–247.

[210] Thomas P Hopp and Kenneth R Woods. "Prediction of protein antigenic determinants from amino acid sequences". In: *Proc. Natl. Acad. Sci. U. S. A.* 78.6 (1981), pp. 3824–3828.

[211] Sujun Hua and Zhirong Sun. "A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach". In: *Journal of molecular biology* 308.2 (2001), pp. 397–407.

[212] Lawrence Hunter. "Molecular biology for computer scientists". In: *Artificial intelligence and molecular biology* (1993), pp. 1–46.

[213] Curtis Huttenhower and Human Microbiome Project Consortium. "Structure, function and diversity of the healthy human microbiome." In: *Nature* 486.7402 (2012). ISSN: 1476-4687. DOI: 10.1038/nature11234. arXiv: NIHMS150003. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3564958%7B%5C&%7Dtool=pmcentrez%7B%5C&%7Drendertype=abstract.

[214] Martijn Huynen et al. "Predicting protein function by genomic context: quantitative evaluation and qualitative inferences". In: *Genome research* 10.8 (2000), pp. 1204–1210.

[215] Rebecca Hwa et al. "Bootstrapping parsers via syntactic projection across parallel texts". In: *Natural language engineering* 11.03 (2005), pp. 311–325.

[216] Nancy Ide. "Cross-lingual sense determination: Can it work?" In: *Computers and the Humanities* 34.1 (2000), pp. 223–234.

[217] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[218] Pavel Ircing et al. “On large vocabulary continuous speech recognition of highly inflectional language-Czech”. In: *Proceedings of the 7th European Conference on Speech Communication and Technology*. Vol. 1. ISCA: International Speech Communication Association. 2001, pp. 487–490.

[219] SM Ashiqul Islam et al. “Protein classification using modified n-grams and skip-grams”. In: *Bioinformatics* (2017), pp. 1481–1487.

[220] Ray Jackendoff. *Semantic Interpretation in Generative Grammar*. 1972. URL: http://eric.ed.gov/ERICWebPortal/recordDetail?accno=ED082548.

[221] Aaron Jaech et al. “Hierarchical Character-Word Models for Language Identification”. In: *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*. 2016, pp. 84–93.

[222] Sabrina Jaeger, Simone Fulle, and Samo Turk. “Mol2vec: Unsupervised machine learning approach with chemical intuition”. In: *J. Chem. Inf. Model.* 58.1 (2018), pp. 27–35.

[223] T Jamali et al. “Nuclear pore complex: biochemistry and biophysics of nucleocytoplasmic transport in health and disease”. In: *Int Rev Cell Mol Biol* 287 (2011), pp. 233–286.

[224] J. Michael Janda and Sharon L. Abbott. *16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: Pluses, perils, and pitfalls*. 2007. DOI: 10.1128/JCM.01228-07.

[225] Alexander Rosenberg Johansen et al. “Deep recurrent conditional random field network for protein secondary prediction”. In: *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM. 2017, pp. 73–78.

[226] Stephen C Johnson. “Hierarchical clustering schemes”. In: *Psychometrika* 32.3 (1967), pp. 241–254.

[227] I T Jolliffe and I T Jolliffe. *Principal Component Analysis*. 1986. URL: http://www.google.com/search?client=safari%7B%5C%%7D7B%7B%5C&%7D%7B%5C%%7D7Drls=en-us%7B%5C%%7D7B%7B%5C&%7D%7B%5C%%7D7Dq=Principal+Component+Analysis%7B%5C%%7D7B%7B%5C&%7D%7B%5C%%7D7Die=UTF-8%7B%5C%%7D7B%7B%5C&%7D%7B%5C%%7D7Doe=UTF-8.

[228] Arttu Jolma et al. “DNA-binding specificities of human transcription factors”. In: *Cell* 152.1-2 (2013), pp. 327–339.

[229] David T Jones. “Protein secondary structure prediction based on position-specific scoring matrices”. In: *Journal of molecular biology* 292.2 (1999), pp. 195–202.

[230] P Jorth et al. "Metatranscriptomics of the Human Oral Microbiome during Health and Disease". In: *mBio* 5.2 (2014), e01012–14–e01012–14. ISSN: 2150-7511. DOI: 10.1128/mBio.01012-14. URL: http://mbio.asm.org/content/5/2/e01012-14.full.pdf+html%7B%5C%%7D5Cnpapers2://publication/doi/10.1128/mBio.01012-14.

[231] Dame Jovanoski, Veno Pachovski, and Preslav Nakov. "On the impact of seed words on sentiment polarity lexicon induction". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016, pp. 1557–1567.

[232] Florence Jungo and Amos Bairoch. "Tox-Prot, the toxin protein annotation program of the Swiss-Prot protein knowledgebase". In: *Toxicon* 45.3 (2005), pp. 293–301.

[233] Dan Jurafsky and James H Martin. *Speech and language processing*. Vol. 3. Pearson London, 2014.

[234] Vanessa Isabell Jurtz et al. "An introduction to deep learning on biological sequence data: examples and solutions". In: *Bioinformatics* 33.22 (2017), pp. 3685–3690.

[235] Nal Kalchbrenner et al. "Neural Machine Translation in Linear Time". In: *CoRR* abs/1610.10099 (2016).

[236] Katharina Kann, Ryan Cotterell, and Hinrich Schuetze. "Neural Morphological Analysis: Encoding-Decoding Canonical Segments". In: *Conference on Empirical Methods in Natural Language Processing*. 2016.

[237] Katharina Kann and Hinrich Schuetze. "MED: The LMU System for the SIGMORPHON 2016 Shared Task on Morphological Reinflection". In: *SIGMORPHON Workshop*. 2016.

[238] Katharina Kann and Hinrich Schuetze. "Single-Model Encoder-Decoder with Explicit Morphological Representation for Reinflection". In: *Annual Meeting of the Association for Computational Linguistics*. 2016.

[239] Ronald M Kaplan and Martin Kay. "Regular models of phonological rule systems". In: *Computational Linguistics* 20.3 (1994), pp. 331–378.

[240] Tobias G Kapp et al. "A comprehensive evaluation of the activity and selectivity profile of ligands for RGD-binding integrins". In: *Sci. Rep.* 7 (2017), p. 39805.

[241] Michael D. Kappelman et al. "The Prevalence and Geographic Distribution of Crohn's Disease and Ulcerative Colitis in the United States". In: *Clinical Gastroenterology and Hepatology* 5.12 (2007), pp. 1424–1429. ISSN: 15423565. DOI: 10.1016/j.cgh.2007.07.012.

[242] Kazutaka Katoh and Daron M Standley. "MAFFT multiple sequence alignment software version 7: improvements in performance and usability". In: *Molecular biology and evolution* 30.4 (2013), pp. 772–780.

[243] Jolanta Kawulok and Sebastian Deorowicz. "CoMeta: Classification of metagenomes using k-mers". In: *PLoS ONE* 10.4 (2015). ISSN: 19326203. DOI: 10.1371/journal.pone.0121453.

[244] Abdellali Kelil et al. "Fast and accurate discovery of degenerate linear motifs in protein sequences". In: *PLoS One* 9.9 (2014), e106081.

[245] Kimmo Kettunen, Paul McNamee, and Feza Baskaya. "Using Syllables As Indexing Terms in Full-Text Information Retrieval". In: *Human Language Technologies - The Baltic Perspective - Proceedings of the Fourth International Conference Baltic HLT 2010, Riga, Latvia, October 7-8, 2010*. 2010, pp. 225–232.

[246] Ariane Khaledi et al. "Fighting antimicrobial resistance in Pseudomonas aeruginosa with machine learning-enabled molecular diagnostics". In: *bioRxiv* (2019), p. 643676.

[247] Christopher SG Khoo and Sathik Basha Johnkhan. "Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons". In: *Journal of Information Science* 44.4 (2018), pp. 491–511.

[248] Sunkyu Kim et al. "Mut2Vec: Distributed representation of cancerous mutations". In: *BMC Med. Genomics* 11.2 (2018), p. 33.

[249] Yoon Kim. "Convolutional neural networks for sentence classification". In: *arXiv preprint arXiv:1408.5882* (2014).

[250] Yoon Kim et al. "Character-Aware Neural Language Models". In: *AAAI Conference on Artificial Intelligence*. 2016, pp. 2741–2749.

[251] S Kimura et al. "Induction of experimental periodontitis in mice with Porphyromonas gingivalis-adhered ligatures." In: *Journal Of Periodontology* 71.7 (2000), pp. 1167–1173. ISSN: 0022-3492. DOI: 10.1902/jop.2000.71.7.1167. URL: http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=pubmed%7B%5C&%7Did=10960025%7B%5C&%7Dretmode=ref%7B%5C&%7Dcmd=prlinks%7B%5C%%7D5Cnpapers2://publication/doi/10.1902/jop.2000.71.7.1167.

[252] Diederik P. Kingma and Jimmy Lei Ba. "Adam: a Method for Stochastic Optimization". In: *International Conference on Learning Representations 2015* (2015), pp. 1–15. ISSN: 09252312. DOI: http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503. arXiv: 1412.6980.

[253] Katrin Kirchhoff et al. "Morphology-based language modeling for conversational Arabic speech recognition". In: *Computer Speech & Language* 20.4 (2006), pp. 589–608.

[254] Dan Klein et al. "Named Entity Recognition with Character-level Models". In: *Computational Natural Language Learning*. 2003, pp. 180–183.

[255] Kevin Knight and Jonathan Graehl. "Machine transliteration". In: *Computational Linguistics* 24.4 (1998), pp. 599–612.

[256] Dan Knights, Elizabeth K. Costello, and Rob Knight. *Supervised classification of human microbiota*. 2011. DOI: 10.1111/j.1574-6976.2010.00251.x.

[257] Tom Kocmi and Ondrej Bojar. "SubGram: Extending Skip-Gram Word Representation with Substrings". In: *Text, Speech, and Dialogue: 19th International Conference, TSD 2016, Brno , Czech Republic, September 12-16, 2016, Proceedings*. Ed. by Petr Sojka et al. Springer, 2016, pp. 182–189.

[258] Alexander F. Koeppel and Martin Wu. "Surprisingly extensive mixed phylogenetic and ecological signals among bacterial Operational Taxonomic Units". In: *Nucleic Acids Research* 41.10 (2013), pp. 5175–5188. ISSN: 03051048. DOI: 10.1093/nar/gkt241.

[259] Stefan Kombrink et al. "Recovery of rare words in lecture speech". In: *International Conference on Text, Speech and Dialogue*. Springer. 2010, pp. 330–337.

[260] Maria Koptjevskaja-Tamm, Martine Vanhove, and Peter Koch. "Typological approaches to lexical semantics". In: *Linguistic typology* 11.1 (2007), pp. 159–185.

[261] Wanqiu Kou, Fang Li, and Timothy Baldwin. "Automatic Labelling of Topic Models Using Word Vectors and Letter Trigram Vectors". In: *Asia Information Retrieval Societies Conference (AIRS)*. 2015, pp. 253–264.

[262] Jiri Kramsky. "A quantitative typology of languages". In: *Language and speech* 2.2 (1959), pp. 72–85.

[263] Alfred L Kroeber and C Douglas Chretien. "Quantitative classification of Indo-European languages". In: *Language* 13.2 (1937), pp. 83–103.

[264] Anders Krogh. "Two methods for improving performance of an HMM and their application for gene finding". In: *Center for Biological Sequence Analysis. Phone* 45 (1997), p. 4525.

[265] William H. Kruskal and W. Allen Wallis. "Use of Ranks in One-Criterion Variance Analysis". In: *Journal of the American Statistical Association* 47.260 (1952), pp. 583–621. ISSN: 1537274X. DOI: 10.1080/01621459.1952.10483441. arXiv: NIHMS150003.

[266] Justin Kuczynski et al. "Direct sequencing of the human microbiome readily reveals community differences". In: *Genome Biology* 11.5 (2010), p. 210. DOI: 10.1186/gb-2010-11-5-210. URL: https://doi.org/10.1186/gb-2010-11-5-210.

[267] Taku Kudo. "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates". In: *arXiv preprint arXiv:1804.10959* (2018).

[268] Vivek Kulkarni et al. "Statistically significant detection of linguistic change". In: *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2015, pp. 625–635.

[269] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[270] Anoop Kunchukuttan and Pushpak Bhattacharyya. "Faster decoding for subword level Phrase-based SMT between related languages". In: *arXiv preprint arXiv:1611.00354* (2016).

[271] Victor Kunin et al. “Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates”. In: *Environmental microbiology* 12.1 (2010), pp. 118–123.

[272] Jack Kyte and Russell F Doolittle. “A simple method for displaying the hydropathic character of a protein”. In: *J. Mol. Biol.* 157.1 (1982), pp. 105–132.

[273] Simon Lacoste-Julien et al. “Block-coordinate Frank-Wolfe optimization for structural SVMs”. In: *arXiv preprint arXiv:1207.4747* (2012).

[274] John Lafferty, Andrew McCallum, and Fernando CN Pereira. “Conditional random fields: Probabilistic models for segmenting and labeling sequence data”. In: (2001).

[275] Guillaume Lample et al. “Neural Architectures for Named Entity Recognition”. In: *Conference of the North American Chapter of the Association for Computational Linguistics / Human Language Technologies*. 2016.

[276] Adrien Lardilleux and Yves Lepage. “Sampling-based multilingual alignment”. In: *Recent Advances in Natural Language Processing*. 2009, pp. 214–218.

[277] Blair Lawley and Gerald W. Tannock. *Analysis of 16S rRNA Gene Amplicon Sequences Using the QIIME Software Package*. Vol. 1537. Springer, 2017, pp. 153–163. DOI: 10.1007/978-1-4939-6685-1_9. URL: http://www.ncbi.nlm.nih.gov/pubmed/27924593%7B%5C%%7D0Ahttp://link.springer.com/10.1007/978-1-4939-6685-1%7B%5C_%7D9.

[278] Yann A LeCun et al. “Efficient backprop”. In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[279] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), p. 436.

[280] Ruth G Ledder et al. “Molecular analysis of the subgingival microbiota in health and disease”. In: *Applied and environmental microbiology* 73.2 (2007), pp. 516–523.

[281] Jason Lee, Kyunghyun Cho, and Thomas Hofmann. “Fully Character-Level Neural Machine Translation without Explicit Segmentation”. In: *CoRR* abs/1610.03017 (2016).

[282] Yves Lepage and Etienne Denoual. “Purest ever example-based machine translation: Detailed presentation and assessment”. In: *Machine Translation* 19.3-4 (2005), pp. 251–282.

[283] Vladimir I Levenshtein. “Binary codes capable of correcting deletions, insertions, and reversals”. In: *Soviet Physics Doklady*. Vol. 10. 8. 1966, pp. 707–710.

[284] Omer Levy and Yoav Goldberg. “Neural word embedding as implicit matrix factorization”. In: *Advances in Neural Information Processing systems*. 2014, pp. 2177–2185.

[285] R. E. Ley et al. "Obesity alters gut microbial ecology". In: *Proceedings of the National Academy of Sciences* 102.31 (2005), pp. 11070–11075. ISSN: 0027-8424. DOI: 10.1073/pnas.0504978102. arXiv: 304. URL: http://www.pnas.org/cgi/doi/10.1073/pnas.0504978102.

[286] Re Ley et al. "Microbial ecology: human gut microbes associated with obesity." In: *Nature* 444.7122 (2006), pp. 1022–3. ISSN: 1476-4687. DOI: 10.1038/nature4441021a. arXiv: NIHMS150003. URL: http://europepmc.org/abstract/MED/17183309.

[287] Heng Li et al. "The sequence alignment/map format and SAMtools". In: *Bioinformatics* 25.16 (2009), pp. 2078–2079.

[288] Weizhong Li and Adam Godzik. "Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences". In: *Bioinformatics* 22.13 (2006), pp. 1658–1659. ISSN: 13674803. DOI: 10.1093/bioinformatics/btl158.

[289] Yu Li et al. "DEEPre: Sequence-based enzyme EC number prediction by deep learning". In: *Bioinformatics* 1 (2017), pp. 760–769.

[290] Wang Ling, Chris Dyer, et al. "Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation". In: *Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 1520–1530.

[291] Wang Ling, Isabel Trancoso, et al. "Character-based Neural Machine Translation". In: *CoRR* abs/1511.04586 (2015).

[292] Pierre Lison and JOerg Tiedemann. "Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles". In: *Proceedings of the 10th International Conference on Language Resources and Evaluation*. 2016.

[293] Bingqiang Liu et al. "An algorithmic perspective of de novo *cis*-regulatory motif finding based on ChIP-seq data". In: *Brief. Bioinform.* (2017), bbx026.

[294] Bo Liu et al. "Deep sequencing of the oral microbiome reveals signatures of periodontal disease". In: *PLoS ONE* 7.6 (2012). ISSN: 19326203. DOI: 10.1371/journal.pone.0037919.

[295] HaiTao Liu and Jin Cong. "Language clustering with word co-occurrence networks based on parallel texts". In: *Chinese Science Bulletin* 58.10 (2013), pp. 1139–1144.

[296] Sharid Loaiciga, Thomas Meyer, and Andrei Popescu-Belis. "English-French Verb Phrase Alignment in Europarl for Tense Translation Modeling". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. Reykjavik, Iceland: European Language Resources Association (ELRA), 2014. URL: http://www.aclweb.org/anthology/L14-1205.

[297] Gerton Lunter and Martin Goodson. "Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads". In: *Genome research* 21.6 (2011), pp. 936–939.

[298] Minh-Thang Luong and Christopher D. Manning. “Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models”. In: *Annual Meeting of the Association for Computational Linguistics*. 2016.

[299] Minh-Thang Luong, Richard Socher, and Christopher D. Manning. “Better Word Representations with Recursive Neural Networks for Morphology”. In: *Computational Natural Language Learning*. 2013.

[300] Nicholas M Luscombe, Dov Greenbaum, Mark Gerstein, et al. “What is bioinformatics? A proposed definition and overview of the field”. In: *Methods of information in medicine* 40.4 (2001), pp. 346–358.

[301] Susan V. Lynch and Oluf Pedersen. “The Human Intestinal Microbiome in Health and Disease”. In: *New England Journal of Medicine* 375.24 (2016), pp. 2369–2379. ISSN: 0028-4793. DOI: 10.1056/NEJMra1600266. URL: http://www.nejm.org/doi/10.1056/NEJMra1600266.

[302] Xuezhe Ma and Eduard H. Hovy. “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF”. In: *Annual Meeting of the Association for Computational Linguistics*. 2016.

[303] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of Machine Learning Research* 9.Nov (2008), pp. 2579–2605.

[304] Chaitanya Malaviya, Graham Neubig, and Patrick Littell. “Learning Language Representations for Typology Prediction”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 2529–2535. URL: http://aclweb.org/anthology/D17-1268.

[305] Christopher D Manning and Hinrich Schuetze. *Foundations of statistical natural language processing*. MIT press, 1999.

[306] RN Mantegna et al. “Systematic analysis of coding and noncoding DNA sequences using methods of statistical linguistics”. In: *Physical Review E* 52.3 (1995), p. 2939.

[307] Ana Marasovic and Anette Frank. “Multilingual Modal Sense Classification using a Convolutional Neural Network”. In: *Proceedings of the 1st Workshop on Representation Learning for NLP*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 111–120. DOI: 10.18653/v1/W16-1613. URL: http://www.aclweb.org/anthology/W16-1613.

[308] Guillaume Marçais and Carl Kingsford. “A fast, lock-free approach for efficient parallel counting of occurrences of k-mers”. In: *Bioinformatics* 27.6 (2011), pp. 764–770. ISSN: 13674803. DOI: 10.1093/bioinformatics/btr011. arXiv: 1006.1266v2.

[309] M Marjorie and Janie Rees-Miller. “Language in Social Contexts”. In: *Contemporary Linguistics* (2001), pp. 537–590.

[310] Benjamin J. Marsland, Koshika Yadava, and Laurent P. Nicod. "The airway microbiome and disease". In: *Chest* 144.2 (2013), pp. 632–637. ISSN: 19313543. DOI: 10.1378/chest.12-2854.

[311] Rebeca Martin et al. "The role of metagenomics in understanding the human microbiome in health and disease". In: *Virulence* 5.3 (2014), pp. 413–423.

[312] Thomas Mayer and Michael Cysouw. "Creating a massively parallel bible corpus". In: *Oceania* 135.273 (2014), p. 40.

[313] Daniel McDonald et al. "An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea". In: *ISME Journal* 6.3 (2012), pp. 610–618. ISSN: 17517362. DOI: 10.1038/ismej.2011.139.

[314] Ryan T McDonald et al. "Universal Dependency Annotation for Multilingual Parsing." In: *ACL (2)*. 2013, pp. 92–97.

[315] Tony McEnery and Richard Xiao. "Domains, text types, aspect marking and English-Chinese translation". In: *Languages in Contrast* 2.2 (1999), pp. 211–229.

[316] Emma McGregor. "Proteins and proteomics: A laboratory manual". In: *Journal of Proteome Research* 3.4 (2004), pp. 694–694.

[317] Alice Carolyn McHardy et al. "Accurate phylogenetic classification of variable-length DNA fragments". In: *Nature Methods* 4.1 (2007), pp. 63–72. ISSN: 15487091. DOI: 10.1038/nmeth976.

[318] April McMahon and Robert McMahon. "Finding families: quantitative methods in language classification". In: *Transactions of the Philological Society* 101.1 (2003), pp. 7–55.

[319] Paul McNamee and James Mayfield. "Character N-Gram Tokenization for European Language Text Retrieval". In: *Information Retrieval* 7.1-2 (2004), pp. 73–97.

[320] John H McWhorter. *Defining creole*. Oxford University Press, 2005.

[321] Ahmed M Mehdi et al. "DLocalMotif: A discriminative approach for discovering local motifs in protein sequences". In: *Bioinformatics* 29.1 (2013), pp. 39–46.

[322] Peter Menzel and Anders Krogh. "Kaiju : Fast and sensitive taxonomic classification for metagenomics". In: *bioRxiv* 7 (2015), pp. 1–9. ISSN: 2041-1723. DOI: 10.1101/031229. URL: http://dx.doi.org/10.1038/ncomms11257.

[323] Sonia Michail et al. "Alterations in the gut microbiome of children with severe ulcerative colitis". In: *Inflammatory Bowel Diseases* 18.10 (2012), pp. 1799–1808. ISSN: 10780998. DOI: 10.1002/ibd.22860.

[324] Rada Mihalcea and Vivi Nastase. "Letter Level Learning for Language Independent Diacritics Restoration". In: *Computational Natural Language Learning*. 2002.

[325] Rada Mihalcea and Michel Simard. "Parallel texts". In: *Natural Language Engineering* 11.03 (2005), pp. 239–246.

[326] Tomas Mikolov, Kai Chen, et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[327] Tomas Mikolov, Ilya Sutskever, Kai Chen, et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems*. 2013, pp. 3111–3119.

[328] Tomas Mikolov, Ilya Sutskever, Anoop Deoras, et al. *Subword language modeling with neural networks*. 2012.

[329] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." In: *HLT-NAACL*. 2013, pp. 746–751.

[330] George A Miller. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[331] Mark Mimee et al. "Programming a Human Commensal Bacterium, Bacteroides thetaiotaomicron, to Sense and Respond to Stimuli in the Murine Gut Microbiota". In: *Cell Systems* 1.1 (2015), pp. 62–71. ISSN: 24054720. DOI: 10.1016/j.cels.2015.06.001. arXiv: 15334406.

[332] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. "Deep learning in bioinformatics". In: *Briefings in Bioinformatics* (2016), bbw068. ISSN: 1467-5463. DOI: 10.1093/bib/bbw068. arXiv: 1603.06430. URL: https://academic.oup.com/bib/article-lookup/doi/10.1093/bib/bbw068.

[333] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. "Deep learning in bioinformatics". In: *Brief. Bioinform.* 18.5 (2017), pp. 851–869.

[334] Yasumasa Miyamoto and Kyunghyun Cho. "Gated Word-Character Recurrent Language Model". In: *Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 1992–1997.

[335] Saif Mohammad and Graeme Hirst. "Distributional measures of concept-distance: A task-oriented evaluation". In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2006, pp. 35–43.

[336] Mary Ann Moran. *The global ocean microbiome*. 2015. DOI: 10.1126/science.aac8455.

[337] Andreas C Mueller and Sven Behnke. "PyStruct: learning structured prediction in python". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 2055–2060.

[338] Thomas Mueller, Helmut Schmid, and Hinrich Schuetze. "Efficient Higher-Order CRFs for Morphological Tagging". In: *Conference on Empirical Methods in Natural Language Processing*. 2013, pp. 322–332.

[339] Stephen H Muggleton et al. "Are grammatical representations useful for learning from biological sequence data: a case study". In: *Journal of Computational Biology* 8.5 (2001), pp. 493–521.

[340] Amitabha Mukerjee, Ankit Soni, and Achla M Raina. “Detecting complex predicates in Hindi using POS projection across parallel corpora”. In: *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*. Association for Computational Linguistics. 2006, pp. 28–35.

[341] Alexey G Murzin et al. “SCOP: a structural classification of proteins database for the investigation of sequences and structures”. In: *Journal of molecular biology* 247.4 (1995), pp. 536–540.

[342] John Myhill and Myhill. *Typological discourse analysis: Quantitative approaches to the study of linguistic function*. Blackwell Oxford, 1992.

[343] Jacob T Nearing et al. “Denoising the Denoisers: An independent evaluation of microbiome sequence error-correction methods”. In: *PeerJ PrePrints* (2018).

[344] Nam-Phuong Nguyen et al. “A perspective on 16S rRNA operational taxonomic unit clustering using sequence similarity”. In: *npj Biofilms and Microbiomes* 2.1 (2016). ISSN: 2055-5008. DOI: 10.1038/npjbiofilms.2016.4. URL: http://www.nature.com/articles/npjbiofilms20164.

[345] Lene Nordrum. “Exploring spontaneous-event marking though parallel corpora: Translating English ergative intransitive constructions into Norwegian and Swedish”. In: *Languages in Contrast* 15.2 (2015), pp. 230–250.

[346] Sebastian Nowozin and Christoph H Lampert. “Structured learning and prediction in computer vision”. In: *Foundations and Trends® in Computer Graphics and Vision* 6.3–4 (2011), pp. 185–365.

[347] FJ Och. *Giza++ software*. 2003.

[348] Franz Josef Och and Hermann Ney. “A systematic comparison of various statistical alignment models”. In: *Computational linguistics* 29.1 (2003), pp. 19–51.

[349] Sarah E Ochsenhirt et al. “Effect of RGD secondary structure and the synergy site PHSRN on cell adhesion, spreading and specific integrin engagement”. In: *Biomaterials* 27.20 (2006), pp. 3863–3874.

[350] Randal S. Olson et al. “Data-driven advice for applying machine learning to bioinformatics problems”. In: *Biocomputing 2018*. WORLD SCIENTIFIC, 2017. DOI: 10.1142/9789813235533_0018. URL: http://www.worldscientific.com/doi/abs/10.1142/9789813235533_0018.

[351] Helieh S Oz and David a Puleo. “Animal models for periodontal disease.” In: *Journal of biomedicine & biotechnology* 2011 (2011), p. 754857. ISSN: 1110-7251. DOI: 10.1155/2011/754857. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3038839%7B%5C&%7Dtool=pmcentrez%7B%5C&%7Drendertype=abstract.

[352] S Banu Ozkan et al. “Protein folding by zipping and assembly”. In: *Proceedings of the National Academy of Sciences* 104.29 (2007), pp. 11987–11992.

[353] Hakime Öztürk, Elif Ozkirimli, and Arzucan Özgür. "A novel methodology on distributed representations of proteins using their interacting ligands". In: *arXiv preprint arXiv:1801.10199* (2018).

[354] Sebastian Pado and Mirella Lapata. "Cross-linguistic Projection of Role-Semantic Information". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. 2005. URL: http://www.aclweb.org/anthology/H05-1108.

[355] Andrew J Page et al. "Roary: rapid large-scale prokaryote pan genome analysis". In: *Bioinformatics* 31.22 (2015), pp. 3691–3693.

[356] Carolina Parada et al. "Learning sub-word units for open vocabulary speech recognition". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. 2011, pp. 712–721.

[357] Donovan H Parks and Robert G Beiko. "Measures of phylogenetic differentiation provide robust and complementary insights into microbial communities". In: *The ISME Journal* 7.1 (Aug. 2012), pp. 173–183. DOI: 10.1038/ismej.2012.88. URL: https://doi.org/10.1038/ismej.2012.88.

[358] Donovan H. Parks et al. "STAMP: Statistical analysis of taxonomic and functional profiles". In: *Bioinformatics* 30.21 (2014), pp. 3123–3124. ISSN: 14602059. DOI: 10.1093/bioinformatics/btu494.

[359] Victoria Pascal et al. "A microbial signature for Crohn's disease". In: *Gut* 66.5 (2017), pp. 813–822. ISSN: 14683288. DOI: 10.1136/gutjnl-2016-313235. arXiv: NIHMS150003.

[360] Edoardo Pasolli et al. "Machine Learning Meta-analysis of Large Metagenomic Datasets: Tools and Biological Insights". In: *PLoS Computational Biology* 12.7 (2016). ISSN: 15537358. DOI: 10.1371/journal.pcbi.1004977.

[361] Kaustubh R. Patil et al. *Taxonomic metagenome sequence assignment with structured output models*. 2011. DOI: 10.1038/nmeth0311-191. arXiv: NIHMS150003.

[362] Joseph N. Paulson et al. "Differential abundance analysis for microbial marker-gene surveys". In: *Nature Methods* 10.12 (2013), pp. 1200–1202. ISSN: 15487091. DOI: 10.1038/nmeth.2658. arXiv: NIHMS150003.

[363] Fabian Pedregosa and G Varoquaux. *Scikit-learn: Machine learning in Python*. Vol. 12. ACM, 2011, pp. 2825–2830. DOI: 10.1007/s13398-014-0173-7.2. arXiv: arXiv:1201.0490v2. URL: http://dl.acm.org/citation.cfm?id=2078195.

[364] Ivo Pedruzzi et al. "HAMAP in 2015: updates to the protein family classification and annotation system". In: *Nucleic acids research* 43.D1 (2014), pp. D1064–D1070.

[365] Fuchun Peng et al. "Language independent authorship attribution using character level language models". In: *Conference of the European Chapter of the Association for Computational Linguistics*. 2003, pp. 267–274.

[366] Jeffrey Pennington, Richard Socher, and Christopher Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[367] PJ Perez-Chaparro et al. “Newly identified pathogens associated with periodontitis: a systematic review”. In: *Journal of dental research* 93.9 (2014), pp. 846–858.

[368] Veronica Perez-Rosas, Carmen Banea, and Rada Mihalcea. “Learning Sentiment Lexicons in Spanish.” In: *LREC*. Vol. 12. 2012, p. 73.

[369] Matthew Peters et al. “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://www.aclweb.org/anthology/N18-1202.

[370] Jane Peterson et al. “The NIH Human Microbiome Project”. In: *Genome Research* 19.12 (2009), pp. 2317–2323. ISSN: 10889051. DOI: 10.1101/gr.096651.109.

[371] Eva Pettersson, Beata Megyesi, and Joakim Nivre. “A Multilingual Evaluation of Three Spelling Normalisation Methods for Historical Text”. In: *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*. 2014, pp. 32–41.

[372] Mohadesseh Peyro et al. *Evolutionary conserved sequence features optimizes nucleoporins behavior for cargo transportation through nuclear pore complex*. 2015.

[373] Ameet J. Pinto, Chuanwu Xi, and Lutgarde Raskin. “Bacterial community structure in the drinking water microbiome is governed by filtration processes”. In: *Environmental Science and Technology* 46.16 (2012), pp. 8851–8859. ISSN: 0013936X. DOI: 10.1021/es302042t.

[374] Ari Pirkola. “Morphological typology of languages for IR”. In: *Journal of Documentation* 57.3 (2001), pp. 330–348.

[375] Barbara Plank, Anders Sogaard, and Yoav Goldberg. “Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss”. In: *Annual Meeting of the Association for Computational Linguistics*. 2016.

[376] Alexander Platzer. “Visualization of SNPs with t-SNE”. In: *PLoS ONE* 8.2 (2013). ISSN: 19326203. DOI: 10.1371/journal.pone.0056883.

[377] Edward F Plow, Thomas A Haas, et al. “Ligand binding to integrins”. In: *J. Biol. Chem.* 275.29 (2000), pp. 21785–21788.

[378] Edward F Plow, Michael D Pierschbacher, et al. “The effect of Arg-Gly-Asp-containing peptides on fibrinogen and von Willebrand factor binding to platelets”. In: *Proc. Natl. Acad. Sci. U. S. A.* 82.23 (1985), pp. 8057–8061.

[379] David Polak et al. “Mouse model of experimental periodontitis induced by Porphyromonas gingivalis Fusobacterium nucleatum infection: Bone loss and host response”. In: *Journal of Clinical Periodontology* 36.5 (2009), pp. 406–410. ISSN: 03036979. DOI: 10.1111/j.1600-051X.2009.01393.x.

[380] Jolinda Pollock et al. “The madness of microbiome: Attempting to find consensus “best practice” for 16S microbiome studies”. In: *Applied and Environmental Microbiology* (2018), AEM.02627–17. ISSN: 0099-2240. DOI: 10.1128/AEM.02627-17. URL: http://aem.asm.org/lookup/doi/10.1128/AEM.02627-17.

[381] Boris T Polyak. “Some methods of speeding up the convergence of iteration methods”. In: *USSR Computational Mathematics and Mathematical Physics* 4.5 (1964), pp. 1–17.

[382] Rachel Poretsky et al. “Strengths and limitations of 16S rRNA gene amplicon sequencing in revealing temporal microbial community dynamics”. In: *PLoS ONE* 9.4 (2014). ISSN: 19326203. DOI: 10.1371/journal.pone.0093827. arXiv: 9809069v1 [arXiv:gr-qc].

[383] James B Procter et al. “Visualization of multiple alignments, phylogenies and gene family evolution”. In: *Nature methods* 7 (2010), S16–S25.

[384] Roman Prytuliak. “Recognition of short functional motifs in protein sequences”. PhD thesis. lmu, 2018.

[385] Roman Prytuliak, Friedhelm Pfeiffer, and Bianca Hermine Habermann. “SLALOM, a flexible method for the identification and statistical analysis of overlapping continuous sequence elements in sequence-and time-series data”. In: *BMC bioinformatics* 19.1 (2018), p. 24.

[386] Roman Prytuliak, Michael Volkmer, et al. “HH-MOTiF: de novo detection of short linear motifs in proteins by hidden Markov model comparisons”. In: *Nucleic Acids Res.* (2017), gkx341.

[387] Christian Quast et al. “The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools”. In: *Nucleic Acids Research* 41.D1 (2013). ISSN: 03051048. DOI: 10.1093/nar/gks1219.

[388] A. Ramezani and D. S. Raj. “The Gut Microbiome, Kidney Disease, and Targeted Interventions”. In: *Journal of the American Society of Nephrology* 25.4 (2014). ISSN: 1046-6673. DOI: 10.1681/ASN.2013080905. URL: http://www.jasn.org/cgi/doi/10.1681/ASN.2013080905.

[389] Ravi Ranjan et al. “Analysis of the microbiome: advantages of whole genome shotgun versus 16S amplicon sequencing”. In: *Biochemical and biophysical research communications* 469.4 (2016), pp. 967–977.

[390] Pushpendre Rastogi, Ryan Cotterell, and Jason Eisner. “Weighting Finite-State Transductions With Neural Context”. In: *Conference of the North American Chapter of the Association for Computational Linguistics / Human Language Technologies*. 2016, pp. 623–633.

[391] Adwait Ratnaparkhi et al. “A maximum entropy model for part-of-speech tagging”. In: *Conference on Empirical Methods in Natural Language Processing*. Vol. 1. Philadelphia, USA. 1996, pp. 133–142.

[392] Emma Redhead and Timothy L Bailey. “Discriminative motif discovery in DNA and protein sequences using the DEME algorithm”. In: *BMC Bioinformatics* 8.1 (2007), p. 385.

[393] Nils Reimers and Iryna Gurevych. “Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Copenhagen, Denmark, Sept. 2017, pp. 338–348. URL: http://aclweb.org/anthology/D17-1035.

[394] Philip Resnik. “Exploiting hidden meanings: Using bilingual text for monolingual annotation”. In: *Computational Linguistics and Intelligent Text Processing* (2004), pp. 283–299.

[395] Philip Resnik. “Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language”. In: *J. Artif. Intell. Res.(JAIR)* 11 (1999), pp. 95–130.

[396] Philip Resnik, Mari Broman Olsen, and Mona Diab. “Creating a parallel corpus from the book of 2000 tongues”. In: *Proceedings of the Text Encoding Initiative 10th Anniversary User Conference (TEI-10)*. Citeseer. 1997.

[397] Vanessa K. Ridaura et al. “Gut microbiota from twins discordant for obesity modulate metabolism in mice”. In: *Science* 341.6150 (2013). ISSN: 10959203. DOI: 10.1126/science.1241214.

[398] Jai Ram Rideout et al. “Subsampled open-reference clustering creates consistent, comprehensive OTU definitions and scales to billions of sequences”. In: *PeerJ* 2 (2014), e545. ISSN: 2167-8359. DOI: 10.7717/peerj.545. URL: https://peerj.com/articles/545.

[399] Peter W Rose et al. “The RCSB protein data bank: Integrative view of protein, gene and 3D structural information”. In: *Nucleic Acids Res.* (2016), gkw1000.

[400] Elizabeth M. Ross et al. “Metagenomic Predictions: From Microbiome to Complex Health and Environmental Phenotypes in Humans and Cattle”. In: *PLoS ONE* 8.9 (2013). ISSN: 19326203. DOI: 10.1371/journal.pone.0073056.

[401] Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. “Ultradense Word Embeddings by Orthogonal Transformation”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 767–777. DOI: 10.18653/v1/N16-1091. URL: https://www.aclweb.org/anthology/N16-1091.

[402] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), pp. 533–536. ISSN: 00280836. DOI: 10.1038/323533a0. arXiv: arXiv:1011.1669v3.

[403] Erkki Ruoslahti. "RGD and other recognition sequences for integrins". In: *Annu. Rev. Cell Dev. Biol.* 12.1 (1996), pp. 697–715.

[404] Kim Rutherford et al. "Artemis: sequence visualization and annotation". In: *Bioinformatics* 16.10 (2000), pp. 944–945.

[405] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. "Dynamic routing between capsules". In: *Advances in neural information processing systems*. 2017, pp. 3856–3866.

[406] Hassan Sajjad. "Statistical models for unsupervised, semi-supervised and supervised transliteration mining". In: *Computational Linguistics* (2012).

[407] CR Sankaran, AD Taskar, and PC Ganeshsundaram. "Quantitative Classification of Languages". In: *Bulletin of the Deccan College Research Institute* (1950), pp. 85–111.

[408] Gillian Sankoff. "The grammaticalization of tense and aspect in Tok Pisin and Sranan". In: *Language Variation and Change* 2.03 (1990), pp. 295–312.

[409] Marianne Elina Santaholma. "Grammar sharing techniques for rule-based multilingual NLP systems". In: *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA)* (2007).

[410] Cicero Nogueira dos Santos and Maira Gatti. "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts". In: *International Conference on Computational Linguistics*. 2014, pp. 69–78.

[411] Cicero Nogueira dos Santos and Victor Guimaraes. "Boosting Named Entity Recognition with Neural Character Embeddings". In: *Fifth Named Entity Workshop*. 2015, pp. 25–33.

[412] Cicero Nogueira dos Santos and Bianca Zadrozny. "Learning Character-level Representations for Part-of-Speech Tagging". In: *International Conference on Machine Learning*. 2014, pp. 1818–1826.

[413] Diana Santos. *Translation-based corpus studies: Contrasting English and Portuguese tense and aspect systems*. 50. Rodopi, 2004.

[414] Delphine M. Saulnier et al. "Gastrointestinal microbiome signatures of pediatric patients with irritable bowel syndrome". In: *Gastroenterology* 141.5 (2011), pp. 1782–1791. ISSN: 00165085. DOI: 10.1053/j.gastro.2011.06.072.

[415] Serge Saxonov et al. "EID: the Exon–Intron Database?an exhaustive database of protein-coding intron-containing genes". In: *Nucleic acids research* 28.1 (2000), pp. 185–190.

[416] Jose U. Scher, Andrew Sczesnak, et al. "Expansion of intestinal Prevotella copri correlates with enhanced susceptibility to arthritis". In: *eLife* 2013.2 (2013). ISSN: 2050084X. DOI: 10.7554/eLife.01202.001.

[417] Jose U. Scher, Carles Ubeda, et al. “Periodontal disease and the oral microbiota in new-onset rheumatoid arthritis”. In: *Arthritis and Rheumatism* 64.10 (2012), pp. 3083–3094. ISSN: 00043591. DOI: 10.1002/art.34539. arXiv: NIHMS150003.

[418] Claudia Schillinger et al. “Co-localized or randomly distributed? Pair cross correlation of in vivo grown subgingival biofilm bacteria quantified by digital image analysis”. In: *PLoS One* 7.5 (2012), e37583.

[419] Sebastian Schlafer et al. “Filifactor alocis-involvement in periodontal biofilms”. In: *BMC microbiology* 10.1 (2010), p. 66.

[420] Patrick D. Schloss et al. “Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities”. In: *Applied and Environmental Microbiology* 75.23 (2009), pp. 7537–7541. ISSN: 00992240. DOI: 10.1128/AEM.01541-09.

[421] Sarah E Schmedes et al. “Targeted sequencing of clade-specific markers from skin microbiomes for forensic human identification”. In: *Forensic Science International: Genetics* 32 (2018), pp. 50–61.

[422] Thomas S B Schmidt, João F. Matias Rodrigues, and Christian von Mering. “Ecological Consistency of SSU rRNA-Based Operational Taxonomic Units at a Global Scale”. In: *PLoS Computational Biology* 10.4 (2014). ISSN: 15537358. DOI: 10.1371/journal.pcbi.1003594.

[423] Tobias Schnabel et al. “Evaluation methods for unsupervised word embeddings”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 298–307.

[424] Hinrich Schuetze. “Word Space”. In: *Advances in Neural Information Processing Systems*. 1992, pp. 895–902.

[425] Hinrich Schütze, Heike Adel, and Ehsaneddin Asgari. “Nonsymbolic Text Representation”. In: *arXiv preprint arXiv:1610.00479* (2016).

[426] Holger Schwenk. “Continuous space language models”. In: *Computer Speech & Language* 21.3 (2007), pp. 492–518.

[427] David B Searls. “A primer in macromolecular linguistics”. In: *Biopolymers* 99.3 (2013), pp. 203–217.

[428] David B Searls. “The computational linguistics of biological sequences”. In: *Artificial intelligence and molecular biology* 2 (1993), pp. 47–120.

[429] David B Searls. “The language of genes”. In: *Nature* 420.6912 (2002), p. 211.

[430] Torsten Seemann. “Prokka: rapid prokaryotic genome annotation”. In: *Bioinformatics* 30.14 (2014), pp. 2068–2069.

[431] Nicola Segata, Daniela Boernigen, et al. “PhyloPhlAn is a new method for improved phylogenetic and taxonomic placement of microbes”. In: *Nature communications* 4 (2013), p. 2304.

[432] Nicola Segata, Jacques Izard, et al. “Metagenomic biomarker discovery and explanation”. In: *Genome Biology* 12.6 (2011). ISSN: 14747596. DOI: 10.1186/gb-2011-12-6-r60. arXiv: Segata,Nicola,2011,Metagenomic.

[433] Terrence J Sejnowski and Charles R Rosenberg. “Parallel networks that learn to pronounce English text”. In: *Complex systems* 1.1 (1987), pp. 145–168.

[434] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Neural machine translation of rare words with subword units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. URL: http://www.aclweb.org/anthology/P16-1162.

[435] M. Ali Basha Shaik et al. “Feature-rich sub-lexical language models using a maximum entropy approach for German LVCSR”. In: *Annual Conference of the International Speech Communication Association*. 2013, pp. 3404–3408.

[436] M Ali Basha Shaik et al. “Hybrid Language Models Using Mixed Types of Sub-Lexical Units for Open Vocabulary German LVCSR.” In: *Annual Conference of the International Speech Communication Association*. 2011, pp. 1441–1444.

[437] Claude E Shannon. “Prediction and entropy of printed English”. In: *Bell Labs Technical Journal* 30.1 (1951), pp. 50–64.

[438] Devyani Sharma. “Typological diversity in new Englishes”. In: *English World-Wide* 30.2 (2009), pp. 170–195.

[439] Valery Shepelev and Alexei Fedorov. “Advances in the exon–intron database (EID)”. In: *Briefings in bioinformatics* 7.2 (2006), pp. 178–185.

[440] Ravi U. Sheth et al. *Manipulating Bacterial Communities by in situ Microbiome Engineering*. 2016. DOI: 10.1016/j.tig.2016.01.005. arXiv: 15334406.

[441] Yusuxke Shibata et al. “Byte pair encoding: a text compression scheme that accelerates pattern matching”. In: *Technical Report DOI-TR-161, Department of Informatics,* (1999). URL: https://pdfs.semanticscholar.org/1e94/41bbad598e181896349757b82af42b6a69
pdf.

[442] Megan Sickmeier et al. “DisProt: the database of disordered proteins”. In: *Nucleic acids research* 35.suppl 1 (2007), pp. D786–D793.

[443] Andrew DM Smith. “Dynamic Models of Language Evolution: The Linguistic Perspective”. In: (2016).

[444] SS Socransky et al. “Microbial complexes in subgingival plaque”. In: *Journal of clinical periodontology* 25.2 (1998), pp. 134–144.

[445] Soren Kaae Sonderby and Ole Winther. “Protein Secondary Structure Prediction with Long Short Term Memory Networks”. In: *arXiv preprint arXiv:1412.7828* (2014).

[446] Jae Jung Song. *Linguistic typology: Morphology and syntax*. Routledge, 2014.

[447] Soren Sonnenburg et al. “Accurate splice site prediction using support vector machines”. In: *BMC bioinformatics* 8.Suppl 10 (2007), S7.

[448] Jose Guilherme Camargo de Souza and Constantin Orasan. “Can projected chains in parallel corpora help coreference resolution?” In: *Discourse Anaphora and Anaphor Resolution Colloquium*. Springer. 2011, pp. 59–69.

[449] Henning Sperr, Jan Niehues, and Alex Waibel. “Letter N-Gram-based Input Encoding for Continuous Space Language Models”. In: *Workshop on Continuous Vector Space Models and their Compositionality*. 2013, pp. 30–39.

[450] Kathrin Spreyer and Anette Frank. “Projection-based acquisition of a temporal labeller”. In: *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*. 2008.

[451] Satish M Srinivasan et al. “Mining for class-specific motifs in protein sequence classification”. In: *BMC bioinformatics* 14.1 (2013), p. 96.

[452] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[453] Nitish Srivastava et al. “Dropout: prevent NN from overfitting”. In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. ISSN: 15337928. DOI: 10.1214/12-AOS1000. arXiv: 1102.4807. URL: http://jmlr.org/papers/v15/srivastava14a.html%7B%5C%%7D5Cnhttp://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf.

[454] Rupesh Kumar Srivastava, Klaus Greff, and Juergen Schmidhuber. “Highway Networks”. In: *ICML 2015 Deep Learing Workshop*. 2015.

[455] Efstathios Stamatatos, Nikos Fakotakis, and George Kokkinakis. “Text genre detection using common word frequencies”. In: *Proceedings of the 18th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics. 2000, pp. 808–814.

[456] Alexander Statnikov et al. “A comprehensive evaluation of multicategory classification methods for microbiomic data”. In: *Microbiome* 1.1 (2013), p. 11. ISSN: 2049-2618. DOI: 10.1186/2049-2618-1-11. URL: http://microbiomejournal.biomedcentral.com/articles/10.1186/2049-2618-1-11.

[457] Kyle Strimbu and Jorge A Tavel. “What are biomarkers?” In: *Current Opinion in HIV and AIDS* 5.6 (2010), p. 463.

[458] Kenji Sugase, H Jane Dyson, and Peter E Wright. “Mechanism of coupled folding and binding of an intrinsically disordered protein”. In: *Nature* 447.7147 (2007), pp. 1021–1025.

[459] Shinichi Sunagawa et al. "Structure and function of the global ocean microbiome". In: *Science* 348.6237 (2015). ISSN: 10959203. DOI: 10.1126/science.1261359. arXiv: NIHMS150003.

[460] Ilya Sutskever, James Martens, and Geoffrey E. Hinton. "Generating Text with Recurrent Neural Networks". In: *International Conference on Machine Learning*. 2011, pp. 1017–1024.

[461] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". In: *Advances in Neural Information Processing Systems*. 2014, pp. 3104–3112.

[462] J A K Suykens and J Vandewalle. "Least Squares Support Vector Machine Classifiers". In: *Neural Processing Letters* 9.3 (1999), pp. 293–300. ISSN: 1573-773X. DOI: 10.1023/A:1018628609742. arXiv: 1018628609742. URL: http://dx.doi.org/10.1023/A:1018628609742.

[463] Diana H Taft et al. "Intestinal microbiota of preterm infants differ over time and between hospitals". In: *Microbiome* 2.1 (2014), p. 36.

[464] Ghazaleh Taherzadeh et al. "Sequence-based prediction of protein–carbohydrate binding sites using support vector machines". In: *Journal of chemical information and modeling* 56.10 (2016), pp. 2115–2122.

[465] Duyu Tang et al. "Learning sentiment-specific word embedding for Twitter sentiment classification". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2014, pp. 1555–1565.

[466] Ben Taskar et al. "Learning structured prediction models: A large margin approach". In: *Proceedings of the 22nd international conference on Machine learning*. ACM. 2005, pp. 896–903.

[467] Ricardo Teles et al. "Lessons learned and unlearned in periodontal microbiology". In: *Periodontology 2000* 62.1 (2013), pp. 95–162.

[468] Joerg Tiedemann and Preslav Nakov. "Analyzing the Use of Character-Level Translation with Sparse and Noisy Datasets". In: *Recent Advances in Natural Language Processing, RANLP 2013, 9-11 September, 2013, Hissar, Bulgaria*. 2013, pp. 676–684.

[469] Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. "The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge". In: *Contemporary oncology* 19.1A (2015), A68.

[470] Elizabeth Closs Traugott. "On the expression of spatio-temporal relations in language". In: *Universals of human language* 3 (1978), pp. 369–400.

[471] Peter J. Turnbaugh et al. "A core gut microbiome in obese and lean twins". In: *Nature* 457.7228 (Nov. 2008), pp. 480–484. DOI: 10.1038/nature07540. URL: https://doi.org/10.1038/nature07540.

[472] Peter D Turney, Patrick Pantel, et al. “From frequency to meaning: Vector space models of semantics”. In: *Journal of artificial intelligence research* 37.1 (2010), pp. 141–188.

[473] Rudra Murthy V, Mitesh M. Khapra, and Pushpak Bhattacharyya. “Sharing Network Parameters for Crosslingual Named Entity Recognition”. In: *CoRR* abs/1607.00198 (2016).

[474] L J P Van Der Maaten and G E Hinton. “Visualizing high-dimensional data using t-sne”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605. ISSN: 1532-4435. DOI: 10.1007/s10479-011-0841-3. arXiv: 1307.1662. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=pubmed%7B%5C&%7Dcmd=Retrieve%7B%5C&%7Ddopt=AbstractPlus%7B%5C&%7Dlist%7B%5C_%7Duids=7911431479148734548related:VOiAgwMNy20J.

[475] Dimitra Vergyri et al. “Morphology-based language modeling for arabic speech recognition.” In: *Annual Conference of the International Speech Communication Association*. Vol. 4. 2004, pp. 2245–2248.

[476] Kevin Vervier et al. “Large-scale machine learning for metagenomics sequence classification”. In: *Bioinformatics* 32.7 (2016), pp. 1023–1032. ISSN: 14602059. DOI: 10.1093/bioinformatics/btv683. arXiv: 1505.06915v1.

[477] Kateřina Veselovská and Ondřej Bojar. “Czech SubLex 1.0”. In: *Charles University, Faculty of Mathematics and Physics, Institute of Formal ...* (2013).

[478] Mauno Vihinen, Esa Torkkila, and Pentti Riikonen. “Accuracy of protein flexibility predictions”. In: *Proteins* 19.2 (1994), pp. 141–149.

[479] David Vilar, Jan-T. Peter, and Hermann Ney. “Can We Translate Letters?” In: *Workshop on Statistical Machine Translation*. 2007.

[480] John K Vries and Xiong Liu. “Subfamily specific conservation profiles for proteins based on n-gram patterns”. In: *BMC bioinformatics* 9.1 (2008), p. 72.

[481] Ekaterina Vylomova et al. “Word Representation Models for Morphologically Rich Languages in Neural Machine Translation”. In: *CoRR* abs/1606.04217 (2016).

[482] Bernhard Waelchli. “The consonant template in synchrony and diachrony.” In: *Baltic linguistics* 1 (2010).

[483] Bernhard Waelchli and Michael Cysouw. “Lexical typology through similarity semantics: Toward a semantic map of motion verbs”. In: *Linguistics* 50(3) (2012), pp. 671–710.

[484] Ulli Waltinger. “GermanPolarityClues: A Lexical Resource for German Sentiment Analysis.” In: *LREC*. 2010, pp. 1638–1642.

[485] Fangping Wan and Jianyang Zeng. “Deep learning with feature embedding for compound-protein interaction prediction”. In: *bioRxiv* (2016), p. 086033.

[486] Hao Wang et al. "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle". In: *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics. 2012, pp. 115–120.

[487] Linlin Wang et al. "Morphological Segmentation with Window LSTM Neural Networks". In: *AAAI Conference on Artificial Intelligence*. 2016.

[488] Lin Wang et al. "Structural modulation of the gut microbiota and the relationship with body weight: compared evaluation of liraglutide and saxagliptin treatment". In: *Scientific reports* 6 (2016), p. 33251.

[489] Qiong Wang et al. "Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy". In: *Applied and environmental microbiology* 73.16 (2007), pp. 5261–5267.

[490] Sheng Wang et al. "Protein secondary structure prediction using deep convolutional neural fields". In: *Scientific reports* 6 (2016), p. 18962.

[491] Zhiyong Wang et al. "Protein 8-class secondary structure prediction using conditional neural fields". In: *Proteomics* 11.19 (2011), pp. 3786–3792.

[492] Michael S Waterman, Temple F Smith, and William A Beyer. "Some biological sequence metrics". In: *Adv. Math. (NY)* 20.3 (1976), pp. 367–387.

[493] Paul J Werbos et al. "Backpropagation through time: what it does and how to do it". In: *Proceedings of the IEEE* 78.10 (1990), pp. 1550–1560.

[494] Lindsay J Whaley. *Introduction to typology: the unity and diversity of language*. Sage Publications, 1996.

[495] Janyce Wiebe, Theresa Wilson, and Claire Cardie. "Annotating expressions of opinions and emotions in language". In: *Language resources and evaluation* 39.2-3 (2005), pp. 165–210.

[496] John Wieting et al. "Charagram: Embedding Words and Sentences via Character n-grams". In: *Conference on Empirical Methods in Natural Language Processing*. 2016.

[497] Felix Ming Fai Wong et al. "Quantifying political leaning from tweets, retweets, and retweeters". In: *IEEE transactions on knowledge and data engineering* 28.8 (2016), pp. 2158–2172.

[498] Derrick E Wood and Steven L Salzberg. "Kraken: Ultrafast metagenomic sequence classification using exact alignments". In: *Genome Biol.* 15.3 (2014), R46.

[499] Derrick E. Wood and Steven L. Salzberg. "Kraken: Ultrafast metagenomic sequence classification using exact alignments". In: *Genome Biology* 15.3 (2014). ISSN: 1474760X. DOI: 10.1186/gb-2014-15-3-r46.

[500] Y. Wu et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: *CoRR* abs/1609.08144 (2016).

[501] RZ Xiao and AM McEnery. “A corpus-based approach to tense and aspect in English-Chinese translation.” In: *The 1st International Symposium on Contrastive and Translation Studies between Chinese and English*. 2002.

[502] Yijun Xiao and Kyunghyun Cho. “Efficient Character-level Document Classification by Combining Convolution and Recurrent Layers”. In: *CoRR* abs/1602.00367 (2016).

[503] Laurie Ann Ximenez-Fyvie, Anne D Haffajee, and Sigmund S Socransky. “Microbial composition of supra-and subgingival plaque in subjects with adult periodontitis”. In: *Journal of clinical periodontology* 27.10 (2000), pp. 722–732.

[504] Wei Xu, Xin Liu, and Yihong Gong. “Document clustering based on non-negative matrix factorization”. In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM. 2003, pp. 267–273.

[505] Xilin Xu et al. “MetaDP: a comprehensive web server for disease prediction of 16S rRNA metagenomic datasets”. In: *Biophysics Reports* 2.5-6 (2016), pp. 106–115.

[506] Ying Xu et al. “PhosContext2vec: a distributed representation of residue-level sequence contexts and its application to general and kinase-specific phosphorylation site prediction”. In: *Scientific reports* 8 (2018).

[507] Nianwen Xue, Yuchen Zhang, and Yaqin Yang. “Distant annotation of Chinese tense and modality”. In: *Proceedings of the IWCS 2013 Workshop on Annotation of Modal Meanings in Natural Language (WAMM)*. Potsdam, Germany: Association for Computational Linguistics, 2013, pp. 47–55. URL: http://www.aclweb.org/anthology/W13-0307.

[508] Mark D Yandell and William H Majoros. “Genomics and natural language processing”. In: *Nat. Rev. Genet.* 3.8 (2002), p. 601.

[509] Ching Chia Yang and Wataru Iwasaki. “MetaMetaDB: A database and analytic system for investigating microbial habitability”. In: *PLoS ONE* 9.1 (2014). ISSN: 19326203. DOI: 10.1371/journal.pone.0087126.

[510] Kevin K Yang et al. “Learned protein embeddings for machine learning”. In: *Bioinformatics* 1 (2018), p. 7.

[511] Yuedong Yang et al. “Sixty-five years of the long march in protein secondary structure prediction: the final stretch?” In: *Briefings in bioinformatics* 19.3 (2016), pp. 482–494.

[512] Zhen Yang et al. “A Character-Aware Encoder for Neural Machine Translation”. In: *International Conference on Computational Linguistics*. 2016, pp. 3063–3070.

[513] Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. “Multi-Task Cross-Lingual Sequence Tagging from Scratch”. In: *CoRR* abs/1603.06270 (2016).

[514] D. Yarowsky, G. Ngai, and R. Wicentowski. “Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora”. In: *Proceedings of the First International Conference on Human Language Technology Research*. 2001. URL: http://www.aclweb.org/anthology/H01-1035.

[515] Seok-Hwan Yoon et al. “Introducing EzBioCloud: a taxonomically united database of 16S rRNA gene sequences and whole-genome assemblies”. In: *International journal of systematic and evolutionary microbiology* 67.5 (2017), pp. 1613–1617.

[516] Lei Yu, Jan Buys, and Phil Blunsom. “Online Segment to Segment Neural Transduction”. In: *Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 1307–1316.

[517] Xiang Zhang and Yann LeCun. “Text Understanding from Scratch”. In: *CoRR* abs/1502.01710 (2015).

[518] Xiang Zhang, Junbo Zhao, and Yann LeCun. “Character-level Convolutional Networks for Text Classification”. In: *Advances in Neural Information Processing Systems*. 2015, pp. 649–657.

[519] Yuchen Zhang and Nianwen Xue. “Automatic Inference of the Tense of Chinese Events Using Implicit Linguistic Information”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1902–1911.

[520] Wei Zhao et al. “Investigating capsule networks with dynamic routing for text classification”. In: *arXiv preprint arXiv:1804.00538* (2018).

[521] Hongyi Zhou and Yaoqi Zhou. “SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures”. In: *Bioinformatics* 21.18 (2005), pp. 3615–3621.

[522] Jian Zhou and Olga G. Troyanskaya. “Deep Supervised and Convolutional Generative Stochastic Network for Protein Secondary Structure Prediction”. In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML’14. Beijing, China: JMLR.org, 2014, pp. I-745–I-753. URL: http://dl.acm.org/citation.cfm?id=3044805.3044890.

[523] Jiyun Zhou et al. “CNNH_PSS: protein 8-class secondary structure prediction by convolutional neural network with highway”. In: *BMC bioinformatics* 19.4 (2018), p. 60.

[524] Naihui Zhou et al. “The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens”. In: *bioRxiv* (2019), p. 653105.

[525] Yaoqi Zhou and Martin Karplus. “Interpreting the folding kinetics of helical proteins”. In: *Nature* 401.6751 (1999), p. 400.

[526] Markéta J Zvelebil et al. “Prediction of protein secondary structure and active sites using the alignment of homologous sequences”. In: *Journal of molecular biology* 195.4 (1987), pp. 957–961.