UNIVERSITY OF CALIFORNIA, SAN DIEGO

Computational Mass Spectrometry: Algorithms for Identification of Peptides not Present in Protein Databases

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

in

Bioinformatics and Systems Biology

by

Julio Ng

Committee in charge:

Professor Pavel A. Pevzner, Chair
Professor Pieter C. Dorrestein, Co-Chair
Professor Vineet Bafna
Professor Steve Briggs
Professor William H. Gerwick

2011

The dissertation of Julio Ng is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2011

DEDICATION

To the memory of Grandpa. You have been my source of inspiration to do well in school.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The most influential person in my professional career is Prof. Pavel Pevzver. He is not only an excellent researcher, but one of my best teachers. He has been an infinite source of ideas for my research career. Prof. Pevzner has taught me to think like a scientist and to communicate my thoughts elegantly, both in public presentations and in writing.

I am also thankful to Prof. Vineet Bafna for being such a positive influence in my undergraduate career to do research in Bioinformatics, which led me to choosing the field for my graduate career. Prof. Pieter Dorrestein has also been an essential force in my research career by proposing challenging problems and providing constant feedback on the various stages of software development. Prof. William Gerwick and Prof. Steve Briggs have been great collaborators and I appreciate their serving in my committee.

I thank Tom Siebel and the Siebel Foundation for their generous gift in recognition of my research. I also thank Sangtae Kim, Wei-Ting Liu, Karen Hom, Kyowon Jeong, Stephen Tanner, Shaojie Zhang and Nuno Bandeira for their help in various projects during my graduate career. All students and researchers in my research group have also been very helpful and excellent colleagues. Finally, I want to thank all my collaborators and coauthors of my scientific publications.

Chapter 2, in full, was published as "Algorithm for identification of fusion proteins via mass spectrometry". J. Ng, and P. A. Pevzner. *Journal of Proteome Research*, vol. 7, no. 1, pp. 89-95, 2008. The dissertation author was the primary author of this paper.

Chapter 3 is in preparation for publication as "Blocked Pattern Matching Problem and its Applications in Computational Proteomics". J. Ng, and P. A. Pevzner 2011, in preparation. The dissertation author is the primary author of this paper.

Chapter 4, in full, was published as "Dereplication and de novo sequencing of nonribosomal peptides". J. Ng, N. Bandeira, W.-T. Liu, M. Ghassemian, T. L. Simmons, W. H. Gerwick, R. Linington, P. C. Dorrestein, and P. A. Pevzner. *Nature Methods*, vol. 6, pp. 596-599, 08 2009. Nuno Bandeira and the dissertation

author were the primary authors of this paper.

Chapter 6, in full, was published as "Interpretation of tandem mass spectra obtained from cyclic nonribosomal peptides". W.-T. Liu, J. Ng, D. Meluzzi, N. Bandeira, M. Gutierrez, T. L. Simmons, A. W. Schultz, R. G. Linington, B. S. Moore, W. H. Gerwick, P. A. Pevzner, and P. C. Dorrestein, *Analytical Chemistry*, vol. 81, no. 11, pp. 4200-4209, 2009. Wei-Ting Liu and the dissertation author were the primary authors of this paper.

VITA

2001-2005	B. S. in Computer Science with Specialization in Bioinformatics *summa cum laude*, University of California, San Diego
2005-2011	Ph. D. in Bioinformatics and Systems Biology, University of California, San Diego

PUBLICATIONS

J. Ng, and P. A. Pevzner, "Blocked Pattern Matching Problem and its Applications in Computational Proteomics". In preparation, 2011.

J. Ng, A. Amir, and P. A. Pevzner, "Blocked Pattern Matching Problem and its Applications in Proteomics". Accepted, RECOMB 2011.

W.-T. Liu, Y.-L. Yang, Y. Xu, A. Lamsa, N. M. Haste, J. Y. Yang, J. Ng, D. Gonzalez, C. D. Ellermeier, P. D. Straight, P. A. Pevzner, J. Pogliano, V. Nizet, K. Pogliano, and P. C. Dorrestein, "Imaging mass spectrometry of intraspecies metabolic exchange revealed the cannibalistic factors of Bacillus subtilis," *Proceedings of the National Academy of Sciences*, vol. 107, no. 37, pp. 16286-16290, 2010.

P. N. Leão, A. R. Pereira, W.-T. Liu, J. Ng, P. A. Pevzner, P. C. Dorrestein, G. M. König, V. M. Vasconcelos, and W. H. Gerwick, "Synergistic allelochemicals from a freshwater cyanobacterium," *Proceedings of the National Academy of Sciences*, vol. 107, no. 25, pp. 11183-11188, 2010.

W.-T. Liu, J. Ng, D. Meluzzi, N. Bandeira, M. Gutierrez, T. L. Simmons, A. W. Schultz, R. G. Linington, B. S. Moore, W. H. Gerwick, P. A. Pevzner, and P. C. Dorrestein, "Interpretation of tandem mass spectra obtained from cyclic nonribosomal peptides," *Analytical Chemistry*, vol. 81, no. 11, pp. 4200-4209, 2009.

J. Ng, N. Bandeira, W.-T. Liu, M. Ghassemian, T. L. Simmons, W. H. Gerwick, R. Linington, P. C. Dorrestein, and P. A. Pevzner, "Dereplication and de novo sequencing of nonribosomal peptides," *Nature Methods*, vol. 6, pp. 596-599, 08 2009.

P. Najmabadi, H. Lee, T. Aung, A. Thuya, J. Ng, J. La Clair, and M. D. Burkart, "Grafta: a 3D environment for biomolecular networks," *International Journal of Bioinformatics Research and Applications*, vol. 5, no. 5, pp. 564-569, 2009.

M. Vingron, L. Wong, N. Bandeira, J. Ng, D. Meluzzi, R. Linington, P. Dorrestein, and P. Pevzner, *De Novo Sequencing of Nonribosomal Peptides*, vol. 4955, pp. 181-195. Springer Berlin / Heidelberg, 2008.

P. A. Pevzner, S. Kim, and J. Ng, "Comment on 'Protein sequences from mastodon and Tyrannosaurus rex revealed by mass spectrometry'," *Science*, vol. 321, no. 5892, pp. 1040, 2008.

J. Ng, and P. A. Pevzner, "Algorithm for identification of fusion proteins via mass spectrometry," *Journal of Proteome Research*, vol. 7, no. 1, pp. 89-95, 2008.

S. Tanner, Z. Shen, J. Ng, L. Florea, R. Guigo, S. P. Briggs, and V. Bafna, "Improving gene annotation using peptide mass spectrometry," *Genome Research*, vol. 17, no. 2, pp. 231-239, 2007.

ABSTRACT OF THE DISSERTATION

Computational Mass Spectrometry: Algorithms for Identification of Peptides not Present in Protein Databases

by

Julio Ng

Doctor of Philosophy in Bioinformatics and Systems Biology

University of California, San Diego, 2011

Professor Pavel A. Pevzner, Chair
Professor Pieter C. Dorrestein, Co-Chair

Mass spectrometry has revolutionized protein identification in the last decade. Efficient algorithms have been developed to identify peptides that are encoded in protein databases. This work presents novel methods for interpretation of mass spectrometry data on compounds that are not directly encoded in protein databases. One example of compounds that are not in protein databases are non-ribosomal peptides. Because of the specialized machinery that synthesizes these compounds and their unique (often cyclic) structure, traditional database search tools cannot analyze these data. With new algorithmic developments, we show that mass spectrometry can speed up the process of characterization of cyclic pep-

tides which are compounds of great interest in drug discovery. A second class of peptides that are not encoded in protein databases are peptides that are the product of fused proteins. This type of peptides can arise from cancer proteomes, where the peptide spans a fusion point. Again, traditional search tools cannot identify fusion peptides because they are not directly encoded in the databases. We also present an algorithm to identify such peptides. Finally, mutated and modified peptides are also not directly encoded in protein databases. Existing tools are particularly inefficient when searching for unexpected modifications. Although this problem has been addressed with "blind" database search tools, their running time is too demanding to be practical for large databases. We develop new methods to search for mutated and modified peptides that are orders of magnitude faster than existing tools. Overall, with new algorithmic developments we enable mass spectrometry to characterize novel compounds that evade identification with traditional MS/MS tools.

Chapter 1

Introduction

Mass spectrometry has been the method of choice for study of proteins in a high-throughput manner. Advances in instrumentation and software has allowed researchers to take over more ambitious projects, both in terms of scale and complexity of the experiments. Mass spectrometry's versatility lies in the fact that it can detect molecules in small concentrations in a wide range of masses. Mass spectrometry has been utilized to detect biomarkers for diseases, identification and quantification of expressed proteins, identification of modifications on proteins, and aiding gene annotations, just to mention a few application in the biological sciences.

1.1 Beyond regular database searches

Because of the ever expanding applications of mass spectrometry, there is also a need to develop new computational methods to interpret the experimental data. The classical peptide identification problem tries to characterize a mass spectrum by assigning an amino acid sequence to it. The sequence can be derived from a database or *de novo*. MASCOT [1], SEQUEST [2], InsPect [3], etc. are representative tools that use the database search approach for mass spectrum identification. However, these algorithms are not designed for applications where the peptide is not in the database. There are biologically significant events that can alter the primary structure of proteins.

In Chapter 2, an algorithm to identify fusion peptides is presented. Genome rearrangements in cancer cells are one scenario in which proteins which altered primary structure can arise. The use of mass spectrometry to detect genome rearrangements has the advantage that it can localize fusion points (breakpoints) of proteins at the amino acid sequence level, as opposed to competing genomic technologies that can only pinpoint rearrangement events in the kilobase scale. Mass spectrometry can also confirm the expression of proteins resulting from genomic rearrangements. Detection of fusion peptides opens another possibility for mass spectrometry to be used as a technology in the search for tumor biomarkers. A technology that is sensitive and cost-efficient.

Lastly, this algorithm can be adapted to search for splice sites, given sufficient peptide coverage and the genome of the organism of interest. This application is useful for alternative splice site detection and validation.

1.2 Scaling searches to very large databases

With advances in mass spectrometry instrumentation, millions of mass spectra can be generated in single experiments. To overcome this problems, researchers have increased the efficiency of peptide identification tools by clustering spectra [4, 5, 6, 7]. Clustering spectra significantly reduces the number of spectra that needs to be searched against the database by up to 90% [7], without losing any peptide identifications. Spectrum datasets are usually redundant and by clustering similar spectra together, a consensus higher quality spectra can be created.

While spectra clustering can reduce the size of the input dataset, the running time of the database search algorithm still depends on the size of the database. The largest databases can reach a few billion of amino acids (human six-frame translation, conglomerate of bacterial proteomes, etc). As a consequence, even the most efficient tag-based (filtering) database search algorithm, InsPect [3] does not scale well for these databases. In Chapter 3, a novel algorithm is presented that results in database search tool that is faster than InsPect for large databases. The tool builds on the MSGappedDictionary [8] tool, a de novo peptide identification

software based on the breakthrough concepts of Spectral Profiles [9] and Spectral Dictionaries [10]. Contrasting with traditional de novo peptide identification tools, MSGappedDictionary generates *gapped peptides*, a pattern of (integer) masses representing masses of one or multiple amino acids. These patterns have a greater filter efficiency than tags generated by InsPect, and usually have very few matches in the database. Gapped peptides also have much greater accuracy than full de novo sequence reconstructions because mass spectra fragmentation is not uniform along all peptide bonds, making it unrealistic to have accurate full reconstructions. Gapped peptides are, therefore, a more natural and accurate way to represent de novo (partial) reconstructions.

MSGappedDictionary is a de novo peptide identification tool and its runtime is independent of the size of the database, but matching gapped peptides to a database could potentially be a very time-consuming task. In fact, it is not clear how to match these patterns to a database efficiently. The classical pattern matching algorithms deal with the problem of exact matching patterns to a text, making them not applicable to the problem of matching gapped peptides to proteome database. The main goal of Chapter 3 is to develop an algorithm to efficiently match gapped peptides to a database. Furthermore, adaptations of the algorithm are presented for mutation tolerance searching using gapped peptides.

1.3 New applications of mass spectrometry

Mass spectrometry revolutionized proteomics when high-throughput peptide identification algorithms were introduced, replacing traditional manual annotation. Application of mass spectrometry to the identification of natural products, specifically on sequencing of cyclic peptides, is a novel area requiring development of computational tools for the interpretation of experimental data. Chapters 4, 6 and 7 describe developments in applications of mass spectrometry to natural product identification and discovery. The hope is that with the development of these computational tools, this field can benefit the same way proteomics benefit when high-throughput peptide sequencing was introduced. The properties

of high-throughput and sensitivity in the field of natural products, where tedious manual interpretation of Nuclear Magnetic Resonance data is expected, is almost unheard of.

In parallel to algorithms for identification of (linear) peptides, algorithms for identification of cyclic peptides also fall into two categories, de novo sequence reconstruction and database search identification. The parallel even extends further when database search methods are mutation tolerant, just as in linear peptide searches. The strategy of a mutation tolerant search in a database of cyclic peptides is called *comparative dereplication*, and it can save significant efforts in natural product research by quickly identifying compounds previously characterized or their known relatives.

1.4 Software Developments

There are mainly two areas in which computational tools can aid researchers make the most out of mass spectrometry datasets. The first is the development and improvement of algorithms for existing and new problems. In the academic environment, most of the resources are dedicated to the development of the most cutting-edge software that outperforms existing tools that solve the same problem, or to the creation of software that tackles a novel problem. In other words, we are very good at developing the smartest algorithms to solve the most difficult problems. In fact, most of the time, we are only limited by the capabilities of the instruments. The lag between new instrumental developments, and software that can take advantage of the new capabilities, is minimal.

The second area, which receives less attention in academic software tools (as opposed to commercial software), is the application of software engineering principles to the development of these software tools. Because most of the resources are dedicated to creating new algorithms and limited resources are dedicated to the support and polishing of such algorithms, software tools never become ubiquitous in the community, regardless of the novelty and power of the algorithms. For example, user friendliness is usually not a top priority when developing software

in academic settings. As a result, when a tool is released, the documentation describes a long list of command line options and a configuration file with obscure and hidden options. Furthermore, it is assumed that the user's computer has all the dependencies to run the software installed.

To this end, several first steps have been taken to remedy the situation. Our group in particular, now has a user-friendly webserver that allows users to run several of our tools (`http://proteomics.ucsd.edu`) with the interface rendered in a web browser freeing the user from installing software dependencies in their local computers and understanding the multiple configuration options of the software. In Chapter 7, a detailed description of the technologies used to create the website to run the cyclic peptide sequencing tools is provided to illustrate methods for easy job submission, interactive documentation and results representation.

User-friendliness is crucial for the popularity of the software, but developer-friendliness is important for the maintenance and further development of the software. Our group has created a central repository for the latest software tools we developed. An effort led by Sangtae Kim and I, we created a umbrella package called `MS_Java` containing all of our algorithms. The package contains various Application Programming Interfaces (API's) that standardize communication between tools authored by different people. These conventions will also be very useful for future developers that want to use existing tools or want to reuse portions of the current algorithms. Object oriented (OO) design is also used extensively to represent common elements in mass spectrometry (i.e. peptide, database, spectrum, etc).

I believe that applying best software engineering practices will make our tools easily maintainable to future developers, and at the same time providing user-friendly interfaces for running our software.

Chapter 2

Algorithm for Identification of Fusion Proteins via Mass Spectrometry

Identification of fusion proteins has contributed significantly to our understanding of cancer progression, yielding important predictive markers and therapeutic targets. While fusion proteins can be potentially identified by mass spectrometry, all previously found fusion proteins were identified using genomic (rather than Mass Spectrometry) technologies. This lack of MS/MS applications in studies of fusion proteins is caused by the lack of computational tools that are able to interpret mass spectra from peptides covering unknown fusion breakpoints (fusion peptides). Indeed, the number of potential fusion peptides is so large that the existing MS/MS database search tools become impractical even in the case of small genomes. We explore computational approaches to identifying fusion peptides, propose an algorithm for solving the Fusion Peptide Identification Problem, and analyze the performance of this algorithm on simulated data. We further illustrate how this approach can be modified for human exons prediction.

2.1 Introduction

Tumor genomes accumulate a large number of rearrangements, many of which contribute to tumor progression and lead to formation of novel fusion genes resulting in oncogenic activities [11]. Identification of fusion genes has contributed significantly to our understanding of cancer progression, yielding important predictive markers and targets for therapeutic intervention such as BCR-ABL, ERBB2, and TP53 [12]. Indeed, the successful anti-leukemia drug STI-571 was designed to abrogate the aberrant activity of the BCR-ABL fusion protein resulting from the Philadelphia chromosome translocation. The breast cancer therapeutic Herceptin was designed to counteract the activity of the HER2-ERBB2 gene. STI-571 and Herceptin are examples where knowledge of fusion genes translated to therapeutics. We remark that identification of fusion genes using genomic approaches is often time-consuming and expensive. In fact, sequence-based analysis of tumor genomes architectures was nearly impossible, or, at best, extremely laborious until very recently [13]. However, fusion genes can be potentially identified by a single mass spectrum provided there exist computational tools that are able to interpret mass-spectra from fusion peptides, allowing fast and cheap detection of fusion proteins. Therefore, the MS-based computational approaches for the identification of fusion proteins are crucial for the identification of tumor biomarkers.

Rearrangements often exert their oncogenic effect by disruption of genes at rearrangement breakpoints [14]. Understanding of cancer is predicated upon knowledge of the architecture of malignant genomes that accumulate a large number of genome rearrangements during tumorigenesis. While recent studies suggest that the known fusion proteins in solid tumors are likely to represent only a tip of an iceberg [15], SKY and other low-resolution technologies are limited to cytogenetic resolution for the localization of tumor breakpoints. Collins's lab (UCSF) recently developed End Sequence Profiling (ESP) approach that maps genome breakpoints associated with genome rearrangements elucidating the structural organization of tumor genomes [16]. ESP recently enabled development of new computational tools for fine-scale analysis of tumor genomes [17, 18] and increased the resolution of cancer studies by two orders of magnitude (100 Kb vs. 10 Mb).

In the last year, other fine-scale techniques for interrogating cancer genomes emerged, most notably the PET technology [19, 20] and the Solexa array platform. The *transcript ESP (tESP)* [16] technique has been recently proposed as an extension of ESP to the analysis of fusion transcripts. tESP, similarly to ESP, requires time-consuming validation that may be difficult to scale. For example, application of tESP to MCF7 [16] resulted in four validated fusion transcripts.[1] However, it remains unclear whether these four transcripts are being translated.

We remark that the existing genome and transcriptome approaches, while powerful, only generate clues for potential fusion transcripts that need to be further validated by other approaches. For example, both ESP and PET technologies may generate a list of hundreds potential "gene pairs" forming fusion genes (most of them may be computational or experimental artifacts) and this list needs to be further winnowed to elucidate the real fusion genes. For example, different "ESP signatures" can be decoded using combinatorial techniques [17, 18, 16] but time-consuming downstream sequencing and functional studies are necessary to distinguish between signatures generated by fusion transcripts (biologically interesting case) and experimental artifacts like chimeric BAC clones. The development of MS-based methods would complement nucleic acid-based methods because of the ability of MS/MS to distinguish fusion peptides from nontranslated abnormal transcripts and experimental artifacts. MS-based proteomic approaches could facilitate the identification of translocation partners and would be applicable to the analysis of samples that are limited in quantity, such as clinical biopsies.

Breakpoints in some oncogenic fusion genes are highly variable from patient to patient, and subtle differences in the positions of BCR-ABL breakpoints may be crucial for disease phenotype [21]. Moreover, knowing the exact breakpoint position facilitates disease monitoring. For example, in different leukemia patients, the breakpoint happens in different introns thus producing the variable junction between the prefix and suffix of the fusion peptide. From this perspective, MS-based approaches would be a useful complement to the DNA-based approaches for the identification of the exact positions of fusion breakpoints.

[1] It should be noted that only two fusion transcripts were previously known to exist in MCF-7, while the ESP consortium has validated four additional fusion transcripts.

The previous attempts at *in silico* identification of tumor-specific fusion transcripts using EST data proved problematic [22] thus calling for applications of an alternative validation technique like mass spectrometry. While MS/MS is an attractive technology for interrogating cancer proteomes, it is not clear how to design algorithms to find the fusion peptides. Moreover, rearrangements of tumor genomes might result in completely novel tumor-specific proteins that are not even recognizable in human proteome [23].

Although MS/MS is an attractive high-throughput approach to discovery and validation of fusion proteins, the first successful MS-based validation of fusion proteins was reported only in 2006 for anaplastic large cell lymphoma (ALCL) often caused by radiation and chemical agents [24]. ALCL is an aggressive lymphoma harboring chromosomal translocations involving the ALK tyrosine kinase. The most common translocation in ALCL leads to formation of a fusion kinase NPM-ALK that activates signaling pathways resulting in enhanced survival and proliferation of tumor cells. So far, ten other translocations involving the ALK gene have been described in lymphomas and a subset of pediatric tumors [25, 26, 27, 28, 29]. For the cases of expressed fusion genes, mass spectrometry is an excellent technique to detect fusion proteins.

ALK forms chimeric fusions with numerous translocation partners. The NPM-ALK and TPM3-ALK fusion proteins have been identified [24] in a tumor biopsy by finding multiple overlapping peptides bridging the breakpoints in the NPM-ALK and TPM3-ALK fusion proteins. This study demonstrated an ability to validate the *known* oncogenic fusion proteins, and the authors raised the problem of adapting their approach for the identification of *unknown* fusion partners. However, no computational details were given on how it can be accomplished. In this paper, we address the problem of identifying fusion peptides and test the algorithm for fusion peptide identification in a simulation experiment with real spectra and mock databases simulating fusion proteins.

Since we failed to find publicly available MS/MS datasets from tumor genomes with annotated fusion breakpoints, we decided also to test our approach in a different application domain: finding peptides bridging two consecutive (unan-

notated) exons. Indeed, if the position of the intron separating these exons is unknown (or incorrectly annotated) then the problem of finding a peptide bridging two consecutive exons is not unlike the problem of finding a fusion peptide. We demonstrated that MS/MS spectra can be used to find introns even if their positions are unknown (or incorrectly annotated). An approach for analyzing alternative splicing via a combination of EST and MS/MS analysis has been recently developed [30]. While this approach proved to be very valuable it can only be successful if the same exon junction is supported by both EST and MS/MS data. Our approach thus complements [30] by demonstrating that high-quality MS/MS spectra can be used for annotating splicing even in the absence of EST data.

2.2 Methods

2.2.1 Fusion Peptide Identification Problem

A *fusion peptide* is defined as an amino acid sequence $p_i \dots p_{i+k} p_j \dots p_{j+l}$, where the prefix $p_i \dots p_{i+k}$ comes from one protein while the suffix $p_j \dots p_{j+l}$ comes from another protein in the *database*, and the *database* is defined as a long amino acid sequence $p_1 \dots p_n$ from the concatenation of all protein sequences from the proteome of an organism. For simplicity's sake, the identity of each protein is not encoded in this definition of the database, but in practice this information is incorporated in the database to enforce the constraint that the prefix must come from a protein different than that of the suffix. The *Fusion Peptide Identification Problem* (FPIP) is to find a fusion peptide in the database that best matches an experimental spectrum. A brute force approach to solving the FPIP searches through all pairs of proteins and all potential fusion sites in these peptides. It results in $O(n^2)$ running time, making this approach impractical.

We propose a method to solve the FPIP that runs in $O(nd)$ time, where n is the number of amino acids in the database and d is the upper bound for the length of the fusion peptide. In practice, it is reasonable to assume that d is 30 since tryptic peptides longer than 30 amino acids are rare and hard to detect via MS/MS.

For simplicity, we represent an experimental spectrum S with precursor mass m as an m-dimensional vector $\vec{s} = s_1 \dots s_m$ where s_i represents the *score* at mass i (most MS/MS database search and de novo tools convert experimental spectra into its scored version, e.g., PRM score [3] or Dančík score [31]). This representation assumes that the spectra are discretized and all masses are integers (for example, for ion-trap spectra this can be approximated by multiplying all masses by 10 and taking integer parts). A peptide P with parent mass m can be represented in a similar way by a binary vector $\vec{\pi} = \pi_1 \dots \pi_m$, where a 1 is placed at every position corresponding to the mass of a prefix of the peptide. The $score(S, P)$ of a match between the spectrum S and the peptide P is defined as the dot product $\vec{s} \cdot \vec{\pi}$, where $\vec{s}$ and $\vec{\pi}$ are vectors corresponding to S and P (for convenience, we assume that the $score(S, P) = -\infty$ if S and P have different precursor masses). In this framework, the FPIP amounts to finding a fusion peptide P that maximizes $score(S, P)$ for all possible fusion peptides in the database, given S.

2.2.2 FPIP Algorithm

Because computing the score for every possible fusion peptide in the database via the brute force approach is impractical, we propose to compute the score for the prefix and suffix of the fusion peptide separately. Let P be a peptide (in a vector representation $\pi_1 \dots \pi_t$) and S be a spectrum (in a vector representation $s_1 \dots s_m$) such that the parent mass of P is smaller than or equal to the parent mass of S (i.e., $t \leq m$). Define $PrefixScore(S, P) = \sum_{i=1}^{t} s_i \pi_i$ and $SuffixScore(S, P) = \sum_{i=1}^{t} s_{i+(m-t)} \pi_i$. Let $P_{i,k}$ be the peptide with k amino acids after position i in the database. The FPIP can be formally defined as follows:

Input A database $p_1 \dots p_n$ and a spectrum S with parent mass m.

Output A fusion peptide $p_i \dots p_{i+k} p_j \dots p_{j+l}$ maximizing $PrefixScore(S, P_{i,k}) + SuffixScore(S, P_{j,l})$ among all fusion peptides of mass m.

Our algorithm shares some features with the MS-Alignment dynamic programming approach for the Mutated Peptide Identification Problem [32] (MPIP).

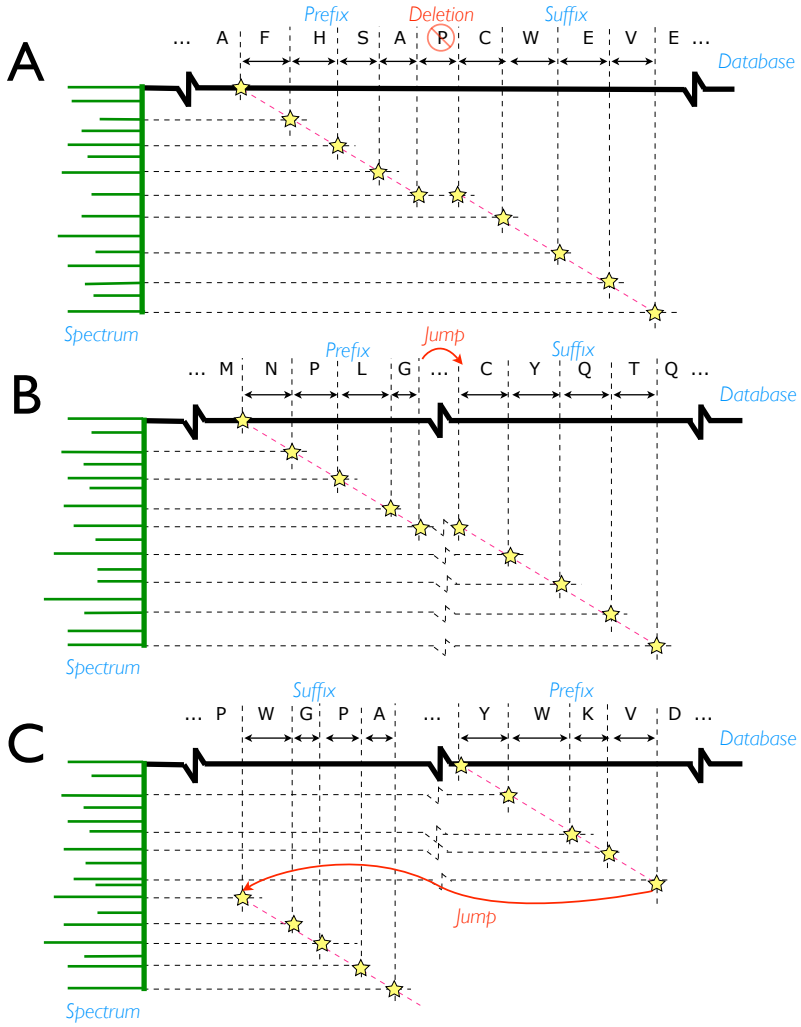


Figure 2.1: Spectral alignment for instances of MPIP and FPIP. For clarity, we only consider b peaks. A) MPIP where the mutation is a deletion of an amino acid in the database separating prefix `FHSA` and suffix `CWEV`. B) FPIP where the prefix `NPLG` comes before the suffix `CYQT` in the database. In this problem, the prefix and suffix come from different proteins in the database. C) Another instance of the FPIP where the prefix `YWKV` comes after the suffix `WGPA` in the database. In the FPIP, the jump could potentially go to any position in the database.

Figure 2.1A illustrates an alignment of a spectrum against a peptide in the database allowing for an internal deletion (an instance of MPIP). A premium is given for each pair of peaks that agree (represented by golden stars in the figure). The peptide that explains the spectrum contains a prefix (before the deletion) and suffix (after the deletion) separated by one amino acid in the protein database. The FPIP (Figure 2.1B) fits perfectly in this prefix-suffix framework if we allow deletions of arbitrary size. Furthermore, it is also possible for the suffix to come before the prefix in the database (Figure 2.1C). This scenario can be modeled by an arbitrary deletion in the spectral alignment. However, our solution to the FPIP is independent of the location of the prefix and suffix in the database.

Given a spectrum S in its vector form $s_1 \dots s_m$ and peptide P as a sequence of amino acids $p_1 \dots p_n$, we formally define:

$$\begin{aligned} \mathit{PrefixScore}(S, P_{i,k}) &= \sum_{j=i}^{i+k} s_{mass(p_i \dots p_j)} \\ \mathit{SuffixScore}(S, P_{i,k}) &= \sum_{j=i}^{i+k} s_{m-mass(p_j \dots p_{i+k})} \end{aligned}$$

where m is the parent mass of S, $0 \leq k \leq d$, $1 \leq i \leq i+k \leq n$, and $mass(p_i \dots p_j) = \sum_{a=i}^{j} mass(p_a)$. We take care of boundary conditions by defining $s_x = -\infty$ for $x < 0$ or $x > m$. $\mathit{PrefixScore}(S, P_{i,k})$ can be computed efficiently for valid values of i and k with the following recurrence:

for $j = 1$ to n **do**
  $\mathit{PrefixMass}(j, 0) = mass(p_j)$
  $\mathit{PrefixScore}(S, P_{j,0}) = s_{mass(p_j)}$
  for $k = 1$ to d **do**
    $\mathit{PrefixMass}(j-k, k) = \mathit{PrefixMass}(j-k, k-1) + mass(p_j)$
    $\mathit{PrefixScore}(S, P_{j-k,k}) = \mathit{PrefixScore}(S, P_{j-k,k-1}) + s_{\mathit{PrefixMass}(j-k,k)}$
  end for
end for

Exceptional cases, $\mathit{PrefixScore}(S, P_{i,k})$ and $\mathit{PrefixMass}(i, k)$ in which the i index becomes 0 or a negative number are not defined. Although these cases are

not handled explicitly in the pseudocode fragment above for simplicity, additional checks are implemented in practice. Similarly, $\mathit{SuffixScore}(S, P_{i,k})$ can be computed efficiently for valid values of i and k with the following recurrence:

1: **for** $j = n$ to 1 **do**
2: $\quad \mathit{SuffixMass}(j, 0) = \mathit{mass}(p_j)$
3: $\quad \mathit{SuffixScore}(S, P_{j,0}) = s_{m-\mathit{mass}(p_j)}$
4: $\quad$ **for** $k = 1$ to d **do**
5: $\quad\quad \mathit{SuffixMass}(j, k) = \mathit{SuffixMass}(j + 1, k - 1) + \mathit{mass}(p_j)$
6: $\quad\quad \mathit{SuffixScore}(S, P_{j,k}) = \mathit{SuffixScore}(S, P_{j+1,k-1}) + s_{m-\mathit{SuffixMass}(j,k)}$
7: $\quad$ **end for**
8: **end for**

Again, for exceptional cases, $\mathit{SuffixScore}(S, P_{i,k})$ and $\mathit{SuffixMass}(i, k)$ where $i + k > n$, these terms are not defined.

We can evaluate all possible candidate fusion prefixes and suffixes with the previous two recurrences in $O(nd)$ time. We also note that the highest scoring fusion peptide must have the highest score prefix and suffix for all possible breakpoints x in spectrum S ($0 \leq x \leq m$). Therefore, this fusion peptide can be found by using two arrays that keep track of the best prefix and suffix (correspondingly) for all possible breakpoints of S. Formally, we define:

$$\mathit{BestPrefixScore}(S, x) = \max_{i,k} \Big(\mathit{PrefixScore}(S, P_{i,k}) \Big)$$
$$\mathit{BestSuffixScore}(S, x) = \max_{i,k} \Big(\mathit{SuffixScore}(S, P_{i,k}) \Big)$$

where $x = \mathit{mass}(p_i \ldots p_{i+k})$, for all $0 \leq x \leq m$ and valid values of i and k. The initial optimization problem can be rewritten as computing:

$$\max_{0 \leq x \leq m} (\mathit{BestPrefixScore}(S, x) + \mathit{BestSuffixScore}(S, m - x))$$

The fusion peptide that best explains S is simply the concatenation of the peptide with the best prefix score to the peptide with the best suffix score.

For our experiments, we chose the Prefix Residue Mass (PRM) spectrum [31, 33] as the vector representation of S and used the *InsPect* [3] implementation to convert raw spectra into PRM spectra.

2.3 Results

2.3.1 Results on Simulated Data

Our test dataset was constructed from a subset of the Human Proteome Organization (HUPO) Plasma Proteome Project (PPP [34]). Specifically, we took the data from Lab 28 which was acquired using an ESI-FTICR instrument. This dataset contained 6 different samples for a combined total of 46297 spectra. 218, 223, 255, 244, 214 and 218 high confidence protein identifications were previously reported in each of the samples [35]. The simulation constructs 2 datasets, the positive set which contains spectra spanning a fusion point, and a negative set which contains spectra not spanning a fusion point.

First we used *InsPect* to search[2] the spectra using the Human IPI Protein Database version 3.34. We construct the positive set from spectra that have a high score. For this experiment, we randomly select 50 spectra that have a Match Quality Score (MQScore [3]) greater than 3.5 and a molecular weight greater than 2000 Daltons for the positive set. The justification for the molecular weight filtering is to allow significant coverage of the prefix and suffix of the fusion spectra. To simulate a fusion event, we remove the proteins that contain the peptide sequences of the spectra in the positive set from the database. We break the proteins at a position covered by each spectrum as illustrated in Figure 2.2. The newly constructed proteins are inserted back into the database. We note that all the peptides identified by the spectra in the positive set are unique and nonoverlapping. The negative set contains 1000 spectra randomly selected from the HUPO PPP dataset that have a molecular weight greater than 2000 Daltons and MQScore no greater than 3.5. The MQScore filtering is applied to discard those spectra that can be readily identified by a regular database search.

The spectra of the positive and negative set were searched against the modified database which is virtually of same size as the original Human IPI Database (over 67,000 entries). The results for the simulation in which the breakpoint is

[2]Search parameters: Tryptic sample, precursor mass tolerance of 0.1 Daltons and no modifications allowed.

simulated to be located in the middle of the peptide (the length of the prefix and suffix are the same) are shown in Figure 2.3.

Additionally, fusion events that are not located in the middle of the peptide were simulated for the positive set. Table 2.1 shows the performance of the fusion peptide algorithm for different positions of the fusion point.

Table 2.1: Results of the simulation experiment for the FPIP. The Breakpoint Offset is defined as the distance from the breakpoint to the midpoint of the peptide. A Breakpoint Offset of 0 indicates the fusion event occurring in the middle of the peptide. The average length of the peptides in the simulations is 20 amino acids.

Breakpoint Offset	Correct Count	Incorrect Count
0	46	4
1	41	9
2	40	10
3	21	29
4	12	38
5	3	47

A single spectrum can be searched against a database with over 28 million amino acids in less than 10 seconds using a modern desktop computer[3]. We note that the brute force approach of constructing a database with all possible fusion events will result in a database in the trillion-amino-acid scale. No database search algorithm today can handle a database of this size.

2.3.2 Splice Site Detection

The described approach for identification of fusion peptides can be adapted for identification of splicing events. We argue that a peptide covering exon junctions can be viewed as a fusion peptide $p_i \dots p_{i+k} p_j \dots p_{j+l}$, where the prefix $p_i \dots p_{i+k}$ and the suffix $p_j \dots p_{j+l}$ come from "close" locations in the six-frame translation of the genome. Although a search against a six-frame translation of the human genome is still not feasible, a search against a six-frame translation of genomic regions of expressed genes can be done.

[3]Apple Power Mac G5 2.5 Ghz with 4GB of RAM

Below we demonstrate that splice sites can be detected given that we have mass spectra bridging two consecutive exons. For simplicity, we only consider splice sites that do not break codons, i.e., starts/ends of exons correspond to starts/ends of codons. A splicing event can be modeled as a deletion of an intron in the database. A spectrum covering a splice site will be explained by a prefix and suffix peptide coming from a three-frame (we know the direction of the gene) translation database (Figure 2.4). The FPIP algorithm was modified so that only "jumps" from different frames or within the same frame of the *same* protein are allowed.

The same spectra from the previous experiment were used for this experiment. The database was constructed by a three-frame translation of 49 genes (exons and introns). The 49 genes were selected because an initial search[4] returned at least two high confidence[5] unique peptides from the protein. The resulting database contained over 29 million amino acids. All spectra from the HUPO PPP were searched against this database and the results are shown in Table 2.2.

2.4 Discussion

While this paper represents the first algorithmic analysis of the fusion peptide identification problem, it falls short of analyzing real tumor samples. This is a reflection of the fact that MS/MS-based studies of fusion proteins started very recently and the spectra resulting from these studies are not publicly available [24].

The current method requires high accuracy data because the optimization routine is very sensitive to parent mass errors. It also requires long peptides that cover a fusion point. We hope that these shortcomings of the algorithm could be overcome by the constant improvement of mass spectrometry instruments. We also note that our simulation used tryptic samples. However, to increase the probability of recovering a peptide that expands a fusion event with a reasonable prefix and suffix, multiple proteases may be used for the sample digestion.

One issue that the current implementation of the algorithm does not address

[4]Search parameters: Tryptic sample, precursor mass tolerance of 0.1 Daltons and no modifications allowed

[5]Filtering criteria: MQScore better than 2.5

Table 2.2: Peptides identified to cover a splice site. The database contained three-frame translations of the genomic regions (exons and introns) of 49 genes. The filtering criteria used for these results are: MQScore $\geq$ 3.5, Precursor Mass $\geq$ 2000 Da and both prefix and suffix must be at least of length 6. We found a total of 19 spectra satisfying these conditions. One spectrum returned the incorrect identification. All results were validated using a regular database search using the real proteins as opposed to the three frame translations of their genomic regions. The count column represents the number of spectra for the given peptide. Protein names with an asterisk (*) next to their name represent proteins in the reverse frame.

Protein	Count	Peptide Prefix Prefix Coordinates	Peptide Suffix Suffix Coordinates
IPI00478003*	3	LHTEAQIQEEGT chr12:9150355-9150390	VVELTGR chr12:9150189-9150209
IPI00017601*	1	LISVDT chr3:150422123-150422140	EHSNIYLQNGPDR chr3:150413136-150413174
IPI00017601*	1	GPEEEHLGILG chr3:150402577-150402609	PVIWAEVGDTIR chr3:150400304-150400339
IPI00022463	3	EDLIWELLNQA chr3:134958508-134958540	QEHFGK chr3:134959300-134959317
IPI00022463	2	SMGGKEDLIWELLNQA chr3:134958493-134958540	QEHFGK chr3:134959300-134959317
IPI00029863	2	WFLLEQPEIQ chr17:1598755-1598784	VAHFPFK chr17:1602630-1602650
IPI00022432	4	ALGISPFHEHAE chr18:27429181-27429216	VVFTANDSGPR chr18:27432529-27432561
IPI00022434	2	NYAEAKDVFLGM chr4:74498181-74498216	FLYEYAR chr4:74499617-74499637

is the event in which the fusion or splicing event breaks a codon. This shortcoming can be solved once the algorithm has access to the genomic data. This implementation of the algorithm does not take genomic sequences as an input. The algorithm can be easily generalized to handle these cases.

Rather than a stand-alone approach to detect fusion events, we believe that this algorithm will be useful in complementing genomic approaches for fusion gene detection. For fusion events that are covered by a peptide, the fusion peptide algorithm can resolve the fusion point within amino-acid resolution.

2.5 Acknowledgements

This chapter, in full, was published as "Algorithm for identification of fusion proteins via mass spectrometry". J. Ng, and P. A. Pevzner. *Journal of Proteome Research*, vol. 7, no. 1, pp. 89-95, 2008. The dissertation author was the primary author of this paper.

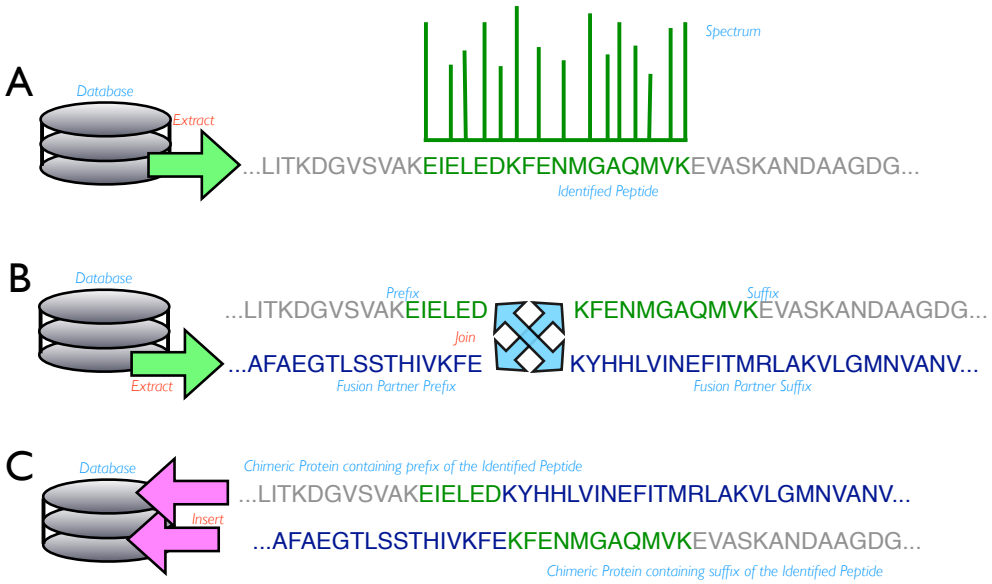


Figure 2.2: Building a mock database simulating fusion peptides: A) The protein entry that explains the given spectrum is extracted from the database. B) This protein is "translocated" with another (randomly picked) protein from the database. C) The translocated (chimeric) protein is inserted back into the database. A "fusion" peptide from the resulting mock database will explain the spectrum.

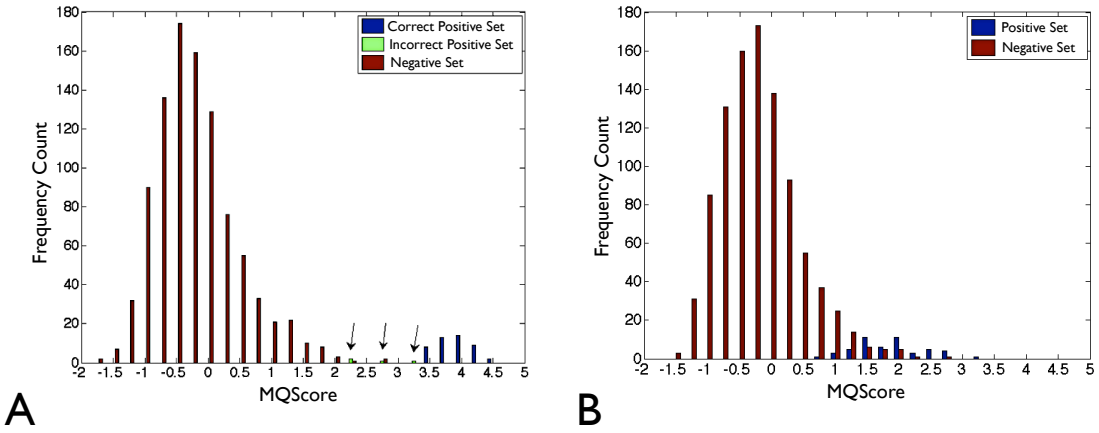


Figure 2.3: MQScore distribution of the Positive and Negative Set. Average length of the peptides in the positive set is 20 amino acids. A) Distribution of scores of the results by the fusion peptide algorithm using a modified Human IPI Database to simulate 50 fusion events covered by 50 spectra. 46 fusion events were recovered successfully (Correct Positive Set) and 4 events were missed (Incorrect Positive Set, indicated by arrows). The Negative set had 1000 spectra, in which 40 spectra did not have any matches at all (no score). B) Distribution of scores of the Positive and Negative set searched against the shuffle version of the previous database. 37 Spectra had no matches in the Negative set.

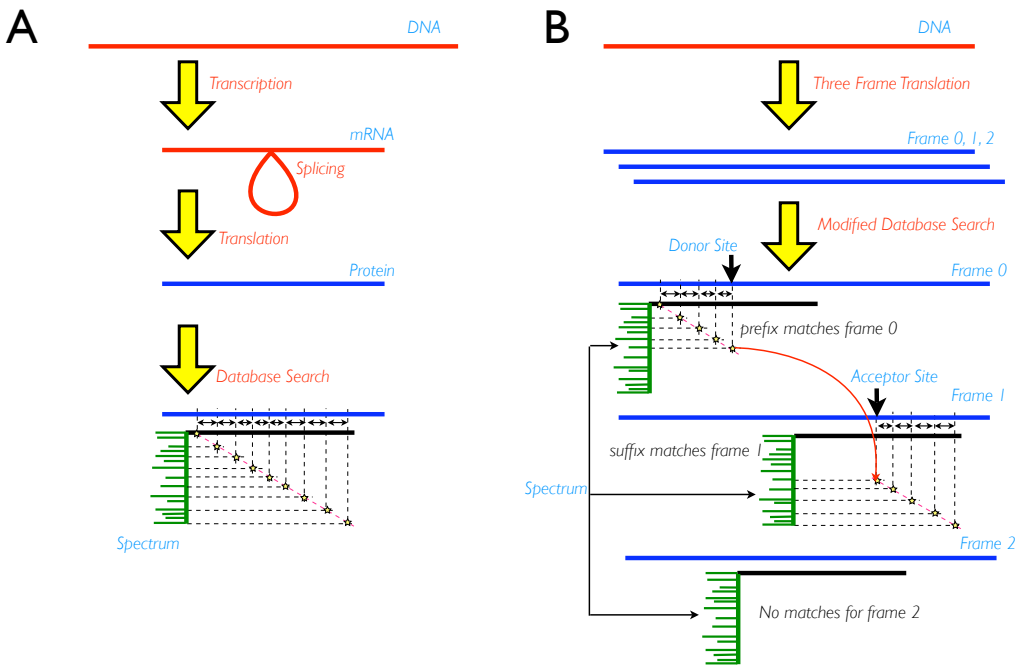


Figure 2.4: Schematic representation of the classical database search problem (A) and our fusion peptide search algorithm applied to splice site detection (B). A splicing event can be modeled as a jump (red arrow) from different translation frames. We only illustrate one scenario in (B), but it is also possible to have a jump in the same frame if the donor and acceptor sites are in the same frame.

Chapter 3

Block Pattern Matching Problem

Matching a *mass spectrum* against a text (a key computational task in proteomics) is slow since the existing text indexing algorithms (with search time independent of the text size) are not applicable in the domain of mass spectrometry. As a result, many important applications (e.g., searches for mutated peptides) are prohibitively time-consuming and even the standard search for non-mutated peptides is becoming too slow with recent advances in high-throughput genomics and proteomics technologies. We introduce a new paradigm – the Blocked Pattern Matching (BPM) - that models peptide identification. BPM corresponds to matching a pattern against a text (over the alphabet of integers) under the assumption that each symbol a in the pattern can match a block of consecutive symbols in the text with total sum a. BPM opens a still unexplored direction in combinatorial pattern matching and leads to the Mutated and Modified BPM (modeling identification of mutated and modified peptides). We illustrate how BPM algorithms solve problems that are beyond the reach of existing proteomics tools.

3.1 Introduction

Matching a tandem mass *spectrum* (MS/MS) to a database is slow as compared to matching a *pattern* to a database. The fundamental algorithmic advantage of the latter approach is that one can *index* the database (e.g., by constructing its suffix tree [36]) so that the complexity of the subsequent queries is not depen-

dent on the database size. Since *efficient* indexing algorithms remain unknown in proteomics[1], many important applications, for example, database searches for mutated peptides, remain extremely time-consuming. Moreover, even the standard applications, such as search for non-modified peptides, are becoming prohibitively slow with recent advances in genomics and proteomics. On one hand, the total size of sequenced bacterial proteomes (most of them are derived from genomes assembled using next generation DNA sequencing technologies) already amounts to billions of amino acids. On the other hand, Ion Mobility Separation (next generation proteomics technology) promises to increase the rate of spectra acquisition by two orders of magnitude.

Tandem mass spectrometry analyzes *peptides* (short 8-30 amino acid long fragments of proteins) by generating their *spectra*. The still unsolved problem in computational proteomics is to reconstruct a peptide from its spectrum: even the advanced *de novo* peptide sequencing tools correctly reconstruct only 30 - 45% of the full-length peptides identified in MS/MS database searches [38, 39]. After two decades of algorithmic developments, it seems that *de novo* peptide sequencing "hits a wall" and that accurate full-length peptide reconstruction is nearly impossible due to the limited information content of spectra.

Recently, with the introduction of MS-GappedDictionary, Jeong et al., 2010 [8] advocated the use of *gapped peptides* to overcome the limitations of full-length *de novo* sequencing. Given a string of n integers $a_1, a_2, \ldots, a_n$ (a peptide) and k integers $1 \leq i_1 < ... < i_k < n$, a *gapped peptide* is a string of $(k+1)$ integers $a_1 + \ldots + a_{i_1}, a_{i_1+1} + \ldots + a_{i_2}, \ldots, a_{i_k+1} + \ldots + a_n$. For example, if a peptide LNRVSQGK is represented as a sequence of its rounded amino acid masses 113, 114, 156, 99, 87, 128, 57, 128 then 113+114, 156+99+87, 128+57, 128 represents a gapped peptide 227, 342, 185, 128. MS-GappedDictionary is a *database filtration* approach based on gapped peptides that are both long and accurate. Gapped peptides have higher accuracy and orders of magnitude higher filtering efficiency

[1]By efficient indexing we mean indexing that typically reduces spectral matching to a single look-up in the indexed database rather than a large number of look-ups. While there is no shortage of useful indexing approaches in proteomics (e.g., fast parent mass indexing like in [37] or peptide sequence tags indexing like in [3]), such approaches may result in a large number of look-ups.

than traditional peptide sequence tags.[2] In contrast to a short peptide sequence tag, a gapped peptide typically has a single match in a proteome, reducing peptide identification to a single database look-up. MS-GappedDictionary generates 25-50 gapped peptides per spectrum (*Pocket Dictionary*) and guarantees that one of them is correct with high probability.

MS-GappedDictionary has a potential to be much faster than traditional proteomics tools because it matches *patterns* rather than *spectra* against a protein database, but algorithms to efficiently match such patterns to a database remain unknown. Therefore, this paper addresses the last missing piece in the series of recent developments aimed at the next generation of peptide identification algorithms [41, 10, 9, 8], an efficient algorithm for matching gapped peptides against a proteome.

Algorithms for matching gapped peptides to a database are unknown because the pattern and the database are encoded in different alphabets displacing the problem outside of the realm of traditional pattern matching. Below we present a fast algorithm for matching gapped peptides against large databases. We demonstrate that with this algorithm, MS-GappedDictionary becomes the fastest protein identification tool, while retaining the accuracy of traditional database search tools. The resulting peptide identification tool is so fast that its running time is dominated by spectral preprocessing rather than scanning the database.[3] In practice, it results in a peptide identification tool that is significantly faster than the state-of-the-art proteomics tools (two orders of magnitude speed-up over Sequest [1] and almost one order of magnitude speed-up over InsPecT [3] in scanning the protein database).

Finally, given the speed advantage of our method, we can now explore other important applications (e.g., searches for mutated peptides and peptides with unanticipated modifications) which were prohibitively time consuming in the past.

[2]While peptide sequence tags are used in many proteomics tools [3, 40], the algorithms for generating *long* sequence tags remain inaccurate. As a result, applications of peptide sequence tags are typically limited to 3 amino acid long tags.

[3]The running time of the existing proteomics algorithms [1, 3] can be partitioned into *spectral preprocessing* time (approximated as $\alpha \cdot \#Spectra$) and database *scanning time* (approximated as $\beta \cdot \#Spectra \cdot ProteomeSize$). For most MS/MS database search tools, the preprocessing time represents a fraction of scanning time.

Particularly, we describe a mutation-tolerant and modification-tolerant (rather than exact) matching of gapped peptides against a database, e.g., detecting mutated and modified peptides. The currently known methods for mutation-tolerant spectral interpretation [32, 42, 43] are slow since they match each spectrum against each peptide in the database. We demonstrate that spectra arising from mutated and modified peptides can be quickly matched to the proteome, providing orders of magnitude speed-up over MS-Alignment [32].

We describe one of many possible applications of the Blocked Pattern Matching (BPM) paradigm when formulated to search for mutations. While traditional MS/MS searches assume that a proteome is known, *proteogenomics* searches use spectra to correct the proteome annotations [44, 45, 46, 43]. The previous proteogenomics approaches searched spectra against the 6-frame translation of the genome in the *standard* genetic code. However, many species use non-standard genetic code [47, 48] that is difficult to establish for a newly sequenced species. In particular, in addition to the standard ATG start codon, GTG and TTG also code for initial methionine (rather than for valine and leucine as in the standard genetic code) in many bacterial genomes. The frequency of non-standard start codons varies widely: in *E. Coli*, GTG and TTG account for 14% and 3% of start codons (not to mention extremely rare ATG and CTG start codons), while in *A. pernix* GTG and TTG are more common than ATG [49]. After a new bacterium is sequenced, the propensities of its starts codons are unknown making accurate gene predictions problematic [50]. Non-standard start codons GTG and TTG (or whatever other) can be discovered by finding mutated peptides in the six-frame translation of a bacterial genome (GTG and TTG correspond to valine and leucine mutated to methionine in the first peptide position). We illustrate applications of this approach to gene annotations in *Anthrobacter* sp.

3.2 Methods

3.2.1 Blocked Pattern Matching Problem

Let $T = t_1, t_2, ..., t_n$ be a *text* over a finite alphabet $\Sigma \subset \mathbb{N}$ and $P = p_1, p_2, ..., p_m$ be a *pattern* over alphabet $\mathbb{N}$ of all natural numbers. Let $S = t_i, t_{i+1}, ..., t_j$ be a substring of T. We define $\overline{S} = \sum_{\ell=i}^{j} t_\ell$ as the *mass* of substring S.

Substrings $t_i, ..., t_j$ and $t_{j+1}, ..., t_k$ are called *consecutive*. A *block* in T is a sequence of consecutive substrings. The *mass* of a block B (denoted $\overline{B}$) is a string comprised of the masses of the consecutive substrings of B. Formally, if $B = S_1 \cdots S_k$ then $\overline{B} = \overline{S_1}, ..., \overline{S_k}$. We say that a pattern P *matches* a text T if there is a block B in T with $\overline{B} = P$.

Example: Let $T = 114, 77, 57, 112, 113, 186, 57, 99, 112, 112, 186, 113, 99$ be a text over an alphabet of 18 symbols that represents masses of 20 amino acids rounded to integers. The consecutive substrings $(57, 112, 113)$, $(186, 57)$, and $(99, 112, 112)$ define a block B in T with $\overline{B} = 282, 243, 323$. Thus, a pattern 282, 243, 323 matches the text T. Below we formulate the Blocked Pattern Matching Problem:

Blocked Pattern Matching (BPM) Problem

Input: A length-n text T and a length-m pattern P.

Output: All blocks B in the text T such that $\overline{B} = P$.

Modern mass spectrometers are capable of producing a million spectra per day and each spectrum corresponds to 25-80 gapped peptides in its Pocket Dictionary [8]. We therefore are interested in matching multiple patterns to the database, and the problem can be formulated as follows:

Multiple Blocked Pattern Matching (MBPM) Problem.

Input. A length-n text T and a set $\mathcal{P}$ of patterns.

Output. All blocks B in the text T such that $\overline{B} = P$, for some $P \in \mathcal{P}$.

The following section presents a step-by-step reduction from a brute force algorithm for the MBPM problem to a more efficient implementation (with search time independent of the text length) that works for relatively short texts T's (a few

million amino acids). The limiting factor for the algorithm that prevents its scaling to longer texts is the exponentially growing memory. To address this bottleneck, we present a practical algorithm that scales to larger proteomes (memory requirements independent of T) at the expense of slower search time.

3.2.2 MBPM algorithms

Brute force MBPM algorithm. Since MS/MS searches typically identify peptides shorter than 30 amino acids, one can limit attention to *k-mers* in the proteome with $k \leq 30$. From the perspective of the BPM problem, we are only interested in patterns that match no more than k symbols in the text. There exist 2^{k-1} ways to break a k-mer into its substrings $b_1, \ldots, b_n$ resulting in 2^{k-1} possible *partitions* $\overline{b_1}, \ldots, \overline{b_n}$ of each k-mer. For example, there exist 8 partitions arising from the 4-mer (114, 77, 99, 57): (114, 77, 99, 57), (114+77, 99, 57), (114, 77+99, 57), (114, 77, 99+57), (114+77,99+57) (114+77+99, 57), (114, 77+99+57), and (114+77+99+57).

Given a set of strings $\mathcal{T}$, we define *KeywordTree*($\mathcal{T}$) as the keyword tree of these strings [36] (See Figure 3.8A for an example of a keyword tree). One can generate all 2^{k-1} partitions for each k-mer in a text T, construct the keyword tree of these partitions, and match each pattern against the constructed keyword tree. This brute force approach can solve the MBPM problem for small proteomes in time independent of the size of the proteome. However, the approach quickly becomes impractical with larger proteomes because the number of partitions of the patterns is large.

Below we describe various improvements over the brute force MBPM algorithm.

MBPM Algorithm: i-unique patterns. In practice, gaps in the gapped peptides have limited sizes, moreover the maximal gap size is a parameter of MS-GappedDictionary [8]. We define a d-bounded pattern as a pattern in the alphabet of integers less than or equal to d. While the number of d-bounded partitions that can be generated from a k-mer is large, below we describe a BPM algorithm that does not require generation of all d-bounded partitions of all k-mers in the text.

Given a position i in a d-bounded pattern $P = p_1, \ldots, p_{i-1}, p_i, p_{i+1}, \ldots, p_n$ and a parameter $1 \leq \delta \leq n-i+1$, (i, δ)-*extension* of P is a pattern $p_1, \ldots, p_{i-1}, p_i + p_{i+1} + \ldots + p_{i+\delta-1}, p_{i+\delta}, \ldots, p_n$ obtained from P by substituting δ symbols in P (starting from the i-th symbol) by their sum. Given a d-bounded pattern P, we define its *extension* $P(i, d)$ as the set of all (i, δ)-extensions of P that result in d-bounded patterns. For example, for a pattern $P = (114, 77, 99, 55, 112)$, $P(2, 300)$ consists of patterns $(114, 77, 99, 55, 112), (114, 176, 55, 112)$ and $(114, 231, 112)$.

A pattern P in the set of patterns $\mathcal{T}$ is called i-*unique* if no other pattern in $\mathcal{T}$ has the same prefix of length i (i-prefix) as P. We now describe a more memory efficient BPM (and MBPM) algorithm (Fig. 3.1). Let $\mathcal{T}_0$ be the set of all k-mers in text T, where each k-mer appears only once. We iteratively construct the set $\mathcal{T}_i$ from the set $\mathcal{T}_{i-1}$ by considering all non-i-unique patterns in $\mathcal{T}_{i-1}$ and substituting each such pattern P by the set of patterns $P(i, d)$. $\mathcal{T}_i$ is the resulting set of patterns with duplicates removed (i.e., each pattern appears only once). The `MBPM` algorithm iteratively generates the sets $\mathcal{T}_1, \ldots, \mathcal{T}_i = \mathcal{T}(T, d)$ and stops when all patterns in the set $\mathcal{T}_i$ become i-unique. We further construct *KeywordTree*($\mathcal{T}(T, d)$) and classify each vertex in this tree as unique or non-unique according to the following rule: Let $q_1, \ldots, q_i$ be a pattern spelled by the path from the root to the vertex v in *KeywordTree*($\mathcal{T}(T, d)$). The vertex v is *unique* if the algorithm `MBPM`$(T, \mathcal{P}, d)$ classified $q_1, \ldots, q_i$ as a prefix of an i-unique pattern at some iteration, and *non-unique* otherwise.

To solve the MBPM Problem, we match each pattern $p_1, \ldots, p_n$ against the constructed keyword tree. In standard searches with the keyword trees, a pattern $p_1, \ldots, p_n$ matches a tree if there exists a path in the tree that spells $p_1, \ldots, p_n$, otherwise the pattern does not match the tree [36]. In contrast, for our application (with special processing of i-unique patterns), failure to find a path that spells $p_1, \ldots, p_n$ does not necessarily implies that the pattern $p_1, \ldots, p_n$ does not match the tree (see `PartitionMatch` function described in Fig. 3.1).

The above algorithm works well for random texts, but deteriorates for texts that have k-mers with long common prefixes (common for real proteomes) because they become i-unique for large i's resulting in a large number of extensions. The

`MBPM`$(T, \mathcal{P}, d)$

1: $\mathcal{T} \leftarrow$ set of all k-mers in T
2: $i \leftarrow 1$
3: **while** there exist non-i-unique patterns in $\mathcal{T}$ **do**
4: remove duplicates from $\mathcal{T}$
5: **for all** non-i-unique pattern $P \in \mathcal{T}$ **do**
6: substitute P by $P(i, d)$ in $\mathcal{T}$
7: **end for**
8: $i \leftarrow i + 1$
9: **end while**
10: construct *KeywordTree*$(\mathcal{T})$
11: **for all** pattern $P \in \mathcal{P}$ **do**
12: `PartitionMatch`$(P,$ *KeywordTree*$(\mathcal{T}))$
13: **end for**

Figure 3.1: `MBPM` algorithm for matching a set of d-bounded patterns $\mathcal{P}$ against the text T (k is an external variable). For the sake of simplicity, the pseudocode hides many details and differences from the actual implementation described in this section. The `PartitionMatch` function works as follows: Let $p_1, \ldots, p_{i-1}$ be the longest prefix of $p_1, \ldots, p_n$ that matches the tree and let v be the last vertex of the path labeled by $p_1, \ldots, p_{i-1}$ in the tree (i.e., no outgoing edge from v is labeled by p_i). If v is a non-unique vertex, we declare that the pattern $p_1, \ldots, p_n$ does not match the the text. Otherwise, we attempt to match the suffix $p_i, \ldots, p_n$ against the path in the keyword tree that start at vertex v. Such matching simply amounts to checking whether the pattern $p_i, \ldots, p_n$ represents a partition of the string spelled by this path. If it is the case, the pattern $p_1, \ldots, p_n$ matches the tree, otherwise there is no match.

next section relaxes the concept of i-uniqueness.

MBPM Algorithm: (i, w)-unique patterns. A pattern P in the set of patterns $\mathcal{P}$ is called (i, w)-*unique* if there are w or less patterns in $\mathcal{P}$ with the same i-prefix as P. The notion of (i, w)-unique patterns generalizes the notion of i-unique patterns (i-unique patterns are $(i, 1)$-unique). We now describe construction of *KeywordTree*($\mathcal{T}(T, d, w)$) that requires less memory than *KeywordTree*($\mathcal{T}(T, d)$). The only difference in constructing *KeywordTree*($\mathcal{T}(T, d, w)$) is that it substitutes the notion of i-unique patterns by the notion of (i, w)-unique patterns in the pseudocode of the *MBPM* algorithm (Fig. 3.1, line 5).

The algorithm iteratively generates the sets $\mathcal{T}_1, \ldots, \mathcal{T}_i = \mathcal{T}(T, d, w)$ until all patterns in the set $\mathcal{T}_i$ become (i, w)-unique. We further construct *KeywordTree*($\mathcal{T}(T, d, w)$) and classify each vertex in this tree as unique or non-unique. Let $q_1, \ldots, q_n$ be a pattern spelled by the path from the root to the vertex v in the keyword tree. The vertex v is *unique* if the *MBPM* algorithm classified $q_1, \ldots, q_n$ as an (i, w)-unique pattern at some iteration, and *non-unique* otherwise. `PartitionMatch` works on the *KeywordTree*($\mathcal{T}(T, d, w)$) structure with a single change. The only difference is when a *unique* vertex is encountered, we will have to follow up to w outgoing paths (rather than a single path) and check whether each of these paths represents a partition of the rest of the unmatched query pattern. We trade off running time for memory in this case.

Figure 3.2 shows the size of $\mathcal{T}(T, d, w)$ for various values of w and illustrates that (i, w)-unique patterns lead to a reasonable memory-speed trade-off. In addition, Figure 3.2A illustrates that for a "random" proteome $\mathcal{T}$ with 100,000 amino acids, it takes only 4 iterations to stabilize $\mathcal{T}(T, 500)$ at ≈ 5 million patterns. The memory requirements increase for real proteomes that typically contain repeats. For example, if a proteome contains two k-mers differing only in the last amino acid, there may be as many as 2^{k-2} patterns generated from these k-mers in the course of the MBPM algorithm with i-unique patterns. As a result, $\mathcal{T}(T, 500)$ for the first 100,000 amino acids of the *Shewanella oneidensis* proteome has over 10 million patterns, a significant increase over a "random" proteome. This increase in total number of sequences for the *Shewanella* proteome as compared to a random

text is due to the fact that the number of k-mers with the same i-prefix in real proteomes is large as compared to a "random" proteome (due to proliferation of repeated segments). This problem can be remedied by using the MBPM on (i, w)-unique patterns with larger w (Fig. 3.2B) leading to a practical solution for the entire *Shewanella* proteome.

Implementing the *KeywordTree*$(\mathcal{T}(T, d))$. We describe an algorithm to construct a data structure equivalent to *KeywordTree*$(\mathcal{T}(T, d))$, without explicitly building the set of patterns $\mathcal{T}(T, d)$. In terms of efficiency, bypassing the set of patterns construction can save computational resources. Furthermore, the concept of i-uniqueness does not need to be explicitly defined when this method is used. The main idea of this algorithm is that first, the keyword tree of all k-mers in T is built and then extensions are done on the nodes of the tree, selectively.

First, we define operations on nodes of the tree so that the final algorithm can be compactly described later on this section. Given a keyword tree of a set of patterns (all k-mers in T), every edge in tree represents a character p_i (in the integer amino acid alphabet) of a pattern P (see Figure 3.6). Some edges might represent a character from multiple patterns, given that their preceding characters (prefixes) are the same.

We define $path(n_i, n_j)$ to be an integer number as follows: if n_j is a child node of n_i, then this is the edge that connect these nodes. if n_j is a descendant node of n_i, then this is the sum of the labels of the edges connecting these nodes. Otherwise, this number is undefined.

Given a node n, $n(d)$ is the set of child nodes, N such that $path(n, n_i) \leq d$ for each n_i in N.

We define `MERGE`$(n_1, ..., n_k)$ as an operation that can be applied to a set on nodes, and that returns a new node with equivalent edges as the outgoing edges of the initial set of merged nodes. The pseudocode for this function is described in Figure 3.3. The main problem when merging multiple nodes is the collision of multiple outgoing edges with the same labels (coming from distinct input nodes). To resolve this problem, we can recursively merge the sink nodes of the colliding edges.

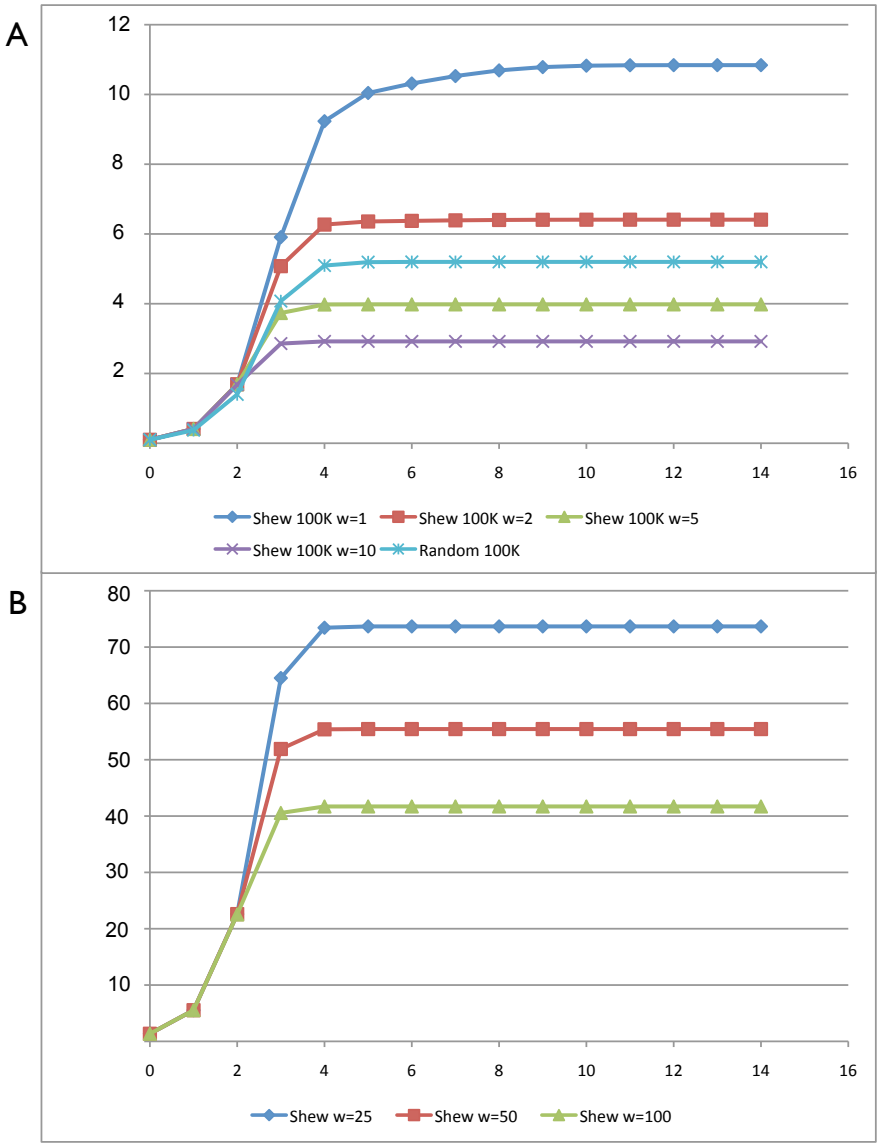


Figure 3.2: Number of distinct patterns generated by the `MBPM` algorithm for various parameters and texts. The y-axis represents the number of patterns (in millions) in the set $\mathcal{T}_i$ generated after the i-th iteration of the $MBPM$ algorithm (x-axis represents the iteration number i). A) Size of $\mathcal{T}_i$ for $i = 0...14$ with $k = 15$ and varying parameter w for the first 100,000 amino acids of the *Shewanella oneidensis* proteome compared to the size of set $\mathcal{T}_i$ for a random sequence of 100,000 amino acids with $w = 1$. The complete proteome of *Shewanella oneidensis* was not used because the number of sequences were too large to fit in memory for $w = 1$. B) Size of $\mathcal{T}_i$ for $i = 0...14$ with $k = 15$ and varying w's for the complete *Shewanella oneidensis* proteome (1.3 millions amino acids).

```
MERGE(n_1, ..., n_k)
 1: E ← ∅
 2: for n_i ∈ n_1, ..., n_k do
 3:    E ←all outgoing edges from n_i
 4: end for
 5: F ← ∅
 6: for E_d ∈ E such that E_d is the set of edges with the label d do
 7:    if |E_d| = 1 then
 8:       F ← E_d
 9:    else
10:       N ← ∅
11:       for e ∈ E_d do
12:          N ← sink(e)
13:       end for
14:       F ← edge(d, MERGE(N))
15:    end if
16: end for
17: return  node(F)
```

Figure 3.3: `MERGE` algorithm for a set of nodes. The function $sink(e)$ returns the sink node of the given edge e. $edge(d, n)$ creates a new edge with label d and with sink node n. $node(E)$ creates a new node with the set E as the outgoing edges.

While the MERGE function operates on a set of nodes, we need to describe the basic extension algorithm on a given node in the tree. EXTEND is the counterpart of the extension algorithm for a pattern P (described previously), resulting in d-bounded patterns. In this case, EXTEND extends all patterns passing through node n at that position. It is clear that if all nodes of the *KeywordTree*($\mathcal{P}$) are extended, all patterns in $\mathcal{P}$ will also be extended. The only trick to complete algorithm is to state the oder in which the nodes should be traversed. Because the EXTEND method creates new nodes and edges on the tree, we have to be careful not to process descendent nodes before the parent node is extended. In order to guarantee this condition, we keep a priority queue giving higher priority to nodes that are closer to the root node. The distance measurement can be simply the label of the path of the current node to the root as defined by the *path* method. The complete algorithm for building a the data structure is described in Figure 3.5. An example, of the transformation algorithm is detailed in Figure 3.6. The last implementation detail is that the resulting data structure can be compressed (analog of a compressed suffix tree [36]) by starting with a compressed keyword tree of the set of patterns, and thus applying EXTEND only to nodes with more than 1 outgoing edge. Essentially, when the EXTEND method is applied to nodes with more than 1 outgoing edge, this is the equivalent of extending non-i-unique patterns. The equivalent optimization for (i, w)-unique patterns is that we EXTEND only on nodes with at least w out degree. The matching algorithm will need to be updated accordingly and it is not described here because it is trivial.

Matching the keyword tree of patterns against the text. While $\mathcal{T}(T, d, w)$ fits into memory for a bacterial proteome, it is infeasible to store this data structure in memory for longer proteomes (e.g., the human six-frame translation). Below we describe a practical solution that scales to proteomes of any size. In the extreme case of trading memory for speed, one can construct the keyword tree of all patterns from the set $\mathcal{P}$ rather than pre-process all k-mers from the text T while solving the MBPM Problem (this amortizes the time for scanning the text). Afterwards, one can match all k-mers from the text against *KeywordTree*($\mathcal{P}$).

Given a set of patterns $\mathcal{P}$, we define $\mathcal{P}_i$ as the set of all i-prefixes (pre-

`EXTEND`(n, d)

1: $E \leftarrow \emptyset$
2: **for** $n_i \in n(d)$ **do**
3: $\quad E \leftarrow path(n, n_i)$
4: **end for**
5: $F \leftarrow \emptyset$
6: **for** $E_d \in E$ such that E_d is the set of edges with the label d **do**
7: $\quad$ **if** $|E_d| = 1$ **then**
8: $\quad\quad F \leftarrow E_d$
9: $\quad$ **else**
10: $\quad\quad N \leftarrow \emptyset$
11: $\quad\quad$ **for** $e \in E_d$ **do**
12: $\quad\quad\quad N \leftarrow sink(e)$
13: $\quad\quad$ **end for**
14: $\quad\quad F \leftarrow edge(d, \texttt{MERGE}(N))$
15: $\quad$ **end if**
16: **end for**
17: replace all outgoing edges of n with F

Figure 3.4: `EXTEND` algorithm for a set of nodes. The function $sink(e)$ returns the sink node of the given edge e. The `MERGE` function is detailed in Figure 3.3.

```
TRANSFORM(P, d)
1: construct KeywordTree(P)
2: add root(KeywordTree(P)) to N
3: while N is not empty do
4:    remove n from N such that path(root(KeywordTree(P)), n) is minimal
5:    n' ← EXTEND(n, d)
6:    for e ∈ edges(n') do
7:       add sink(e) to N
8:    end for
9: end while
```

Figure 3.5: TRANSFORM algorithm to construct a data structure encoding for all the extended patterns of $\mathcal{P}$. The function $edges(n)$ constructs the set of outgoing edges from node n. The function $sink(e)$ returns the sink node of the given edge e. The EXTEND function is detailed in Figure 3.4. The set of patterns $\mathcal{P}$ is actually all k-mers in T.

fixes of length i) of patterns from $\mathcal{P}$. A pattern P (that does not necessarily belongs to $\mathcal{P}$) is called $(i, \mathcal{P})$-*compliant* if the i-prefix of P belongs to $\mathcal{P}_i$. Given a d-bounded pattern P, we define its *extension* $P(i, d, \mathcal{P})$ as the set of all (i, δ)-extensions of P that result in d-bounded *and* $(i, \mathcal{P})$-compliant patterns. For example, for a pattern $P = (114, 77, 99, 55, 112)$ and a set $\mathcal{P}$ consisting of three patterns $(114, 77, 131, 112), (114, 55, 112)$ and $(114, 231, 112, 242, 131)$, $P(2, 300)$ consists of patterns $(114, 77, 99, 55, 112)$, $(114, 176, 55, 112)$ and $(114, 231, 112)$. $P(2, 300, \mathcal{P})$ consists of only $(114, 77, 99, 55, 112)$ and $(114, 231, 112)$ because the pattern $(114, 176, 55, 112)$ is not $(i, \mathcal{P})$-compliant for $i = 2$.

A more memory efficient way to build the set of patterns $\mathcal{T}$ in Figure 3.1 can be achieved by changing line 6: "substitute P by $P(i, d)$ in $\mathcal{T}$" into a line "substitute P by $P(i, d, \mathcal{P})$ in $\mathcal{T}$". We call the final set of patterns $\mathcal{T}(T, d, \mathcal{P})$. In practice, $|P(i, d, \mathcal{P})| \ll |P(i, d)|$, and therefore the $KeywordTree(\mathcal{T}(T, d, \mathcal{P}))$ is much smaller than $KeywordTree(\mathcal{T}(T, d))$. However, all patterns $\mathcal{T}(T, d, \mathcal{P})$ may still not fit into memory when both the text T is very large. In this case, we can break T into shorter segments (so that the data structure for each segment fits

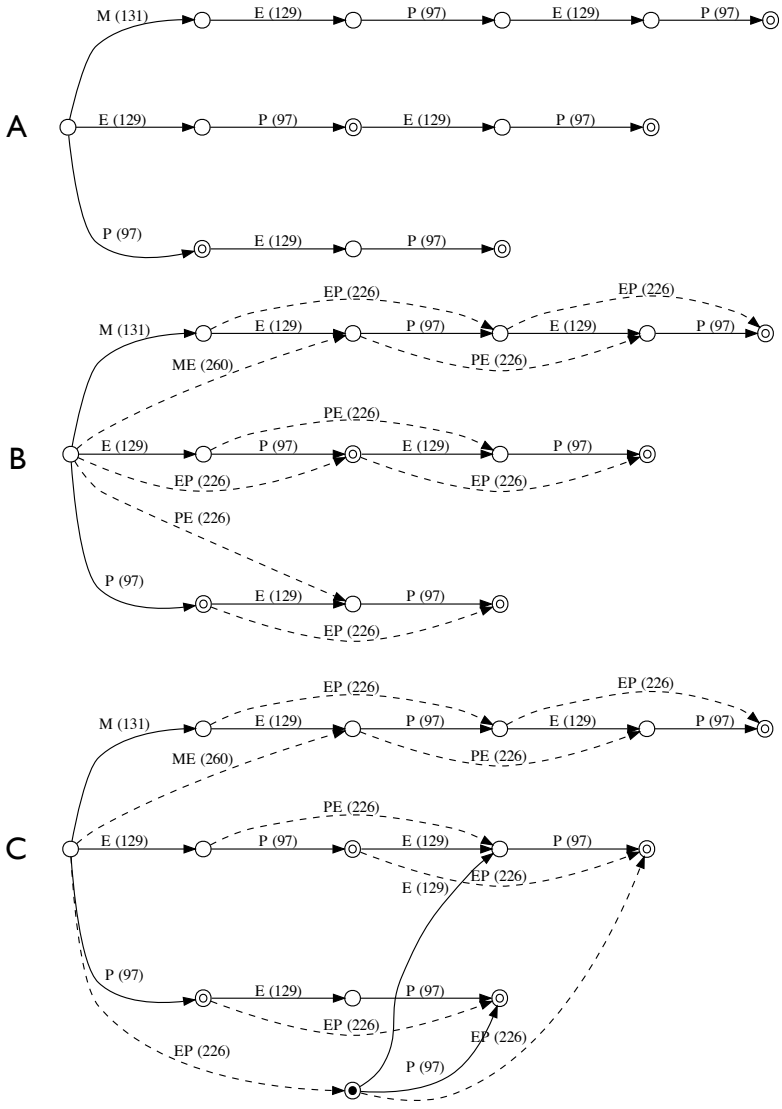


Figure 3.6: Construction of keyword tree of all suffixes of T = MEPEP (suffix trie). A) Suffix trie for T. There are 5 suffixes in the trie ending with nodes with double circles. The labels of the edges are represented as integer masses. B) Addition of edges (dashed) to the trie in order to represent all gap masses of 2 adjacent amino acids. C) Resolving of conflicting edge PE and EP with the same mass outgoing from the root resulting in a new (shaded) node. The changes are propagated to make the trie consistent.

`MBPMopt`$(T, \mathcal{P}, d)$

```
1:  for l = 1 ... |T| do
2:    T ← k-mer at T_l
3:    i ← 1
4:    while T is not empty do
5:      for all P ∈ T do
6:        substitute P by P(i, d, P) in T
7:        if i-prefix of P matches Q ∈ P then
8:          pattern Q matches text T
9:        end if
10:     end for
11:     i ← i + 1
12:   end while
13: end for
```

Figure 3.7: Memory efficient implementation of the MBPM algorithm for matching a set of d-bounded patterns $\mathcal{P}$ against the text T by partitioning T into individual k-mers.

into memory) and solve the MBPM Problem separately for each segment.

In the extreme case, we consider each k-mer $T_i = t_i, t_{i+1} \ldots, t_{i+k-1}$ separately and construct $\mathcal{T}(T_i, d, \mathcal{P})$ (for $1 \leq i \leq n - k + 1$) resulting in a memory efficient albeit slower algorithm. The trade-off of this extreme implementation is that we lose the i-uniqueness optimization because all patterns resulting from extensions of a single k-mer are i-unique. However, as it turns out, the speed and simplicity of this implementation work well in practice.

The pseudocode of the algorithm (`MBPMopt`) is summarized in Figure 3.7. The pseudocode is written such that it resembles Figure 3.1, but at the implementation level many optimizations can be applied and some of the input parameters can be relaxed (see below). Figure 3.8 illustrates the work of the algorithm run for a single k-mer in T.

Optimizations to the `MBPMopt` Algorithm. First, *keywordTree*$(\mathcal{T})$ does not need to be explicitly built in order to match a k-mer to a pattern in $\mathcal{P}$. As

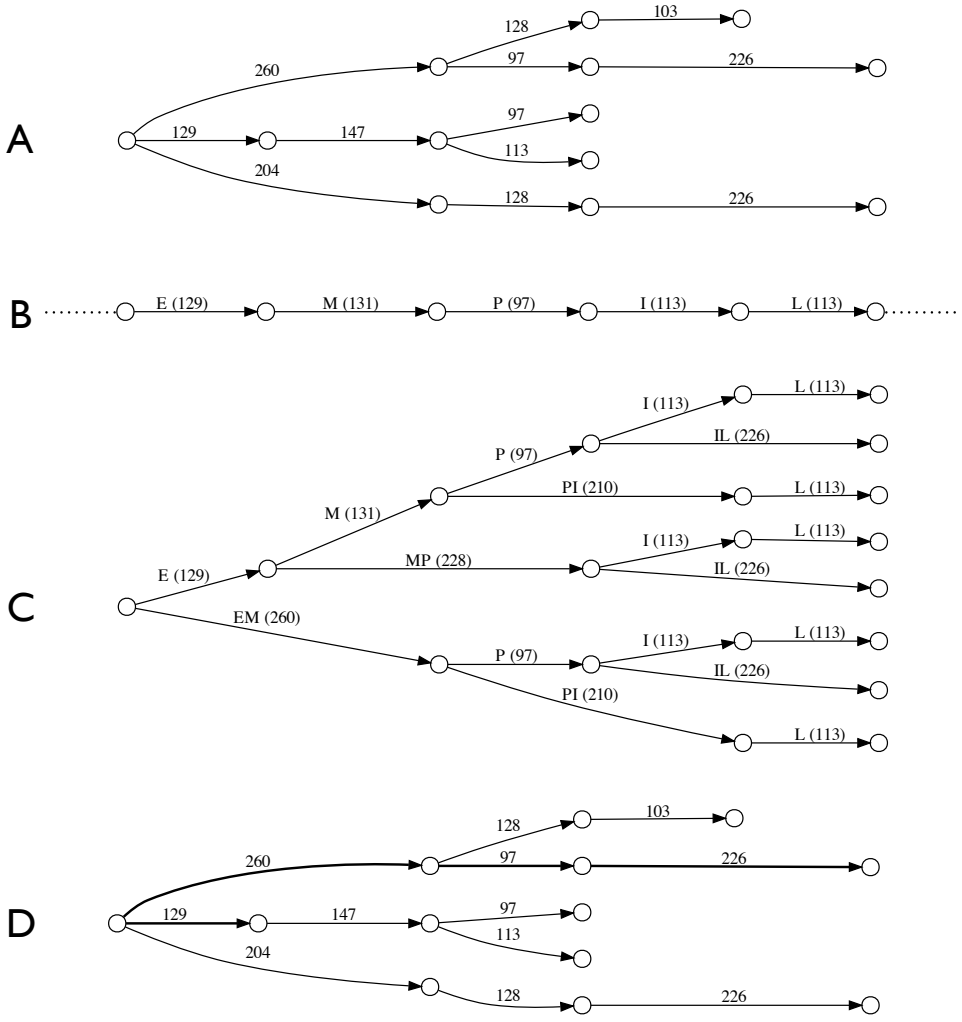


Figure 3.8: Matching a keyword tree of patterns (*KeywordTree*($\mathcal{P}$)) against a k-mer from text T. A) *KeywordTree*($\mathcal{P}$) built for a set $\mathcal{P}$ of 5 patterns: [129, 147, 97], [129, 147, 113], [204, 128, 226], [260, 97, 226] and [260, 128, 103]. B) A text T containing k-mer EMPIL ([129, 131, 97, 113, 113]). C) Construction of *KeywordTree*($\mathcal{T}$(EMPIL, 300)). D) Intersection of *KeywordTree*($\mathcal{T}$(EMPIL, 300, $\mathcal{P}$)) and *KeywordTree*($\mathcal{P}$) shown in bold. The complete *KeywordTree*($\mathcal{P}$) is also shown on the background.

a consequence, `PartitionMatching` does not need to be invoked in the algorithm because as soon as we find i-prefix matching a pattern $Q \in \mathcal{P}$, we know that pattern Q matches a prefix of the current k-mer. Additionally, k does not need to be fixed because it can be dynamically adjusted depending on the value of i when all patterns in $\mathcal{T}$ are non-$(i, \mathcal{P})$-compliant. Similarly, d does not need to be fixed because we can substitute P by $P(i, d, \mathcal{P})$ for any value of d without significant increase in the size of the set of resulting patterns. The result is that the construction of $\mathcal{T}$ is always guided by $\mathcal{P}$.

3.2.3 Mutation-Tolerant Peptide Identification

This section describes an algorithm for mutation-tolerant matching gapped peptides to a database. Let $T_i(a)$ be a text obtained from a text $T = t_1, \ldots t_n$ by substituting a symbol a instead of the i-th symbol of T (for $1 \leq i \leq n$ and $a \in \Sigma$ where Σ is the set of amino acid masses). For example, if $T = 57, 112, 113, 113, 186$, $T_3(99) = 57, 112, 99, 113, 186$. To accommodate for insertions and deletions, we denote $T_i(\emptyset)$ as the deletion of the i-th symbol of T and $T_i(a^+)$ as the insertion of a symbol a before the i-th symbol of T. For example, $T_3(\emptyset) = 57, 112, 113, 186$ and $T_3(128^+) = 57, 112, 128, 113, 113, 186$. A *mutated block* in the text T is a block in $T_i(a)$ (for some i and a). For example, substrings (112,128) and (113,113,186) form a mutated block (240,412) in the text $T = 57, 112, 113, 113, 186$ because they form a block in $T_3(128^+)$. We are interested in solving the following problem:

Mutated Multiple Blocked Pattern Matching (MutMBPM) Problem

Input. A text T over Σ and a set $\mathcal{P}$ of patterns.

Output. All mutated blocks B in the text T such that $\overline{B} = P$, for some $P \in \mathcal{P}$.

While the MutMBPM Problem is limited to peptide identified with a *single* mutations, this is a reasonable limitation in practice. Indeed, Single Amino Acid Polymorphisms (SAAPs) are rarely clustered in the same region of proteins implying that the identified peptides (that are typically shorter than 30 amino acids) are unlikely to have more than one mutation. Also, the false discovery rate in

searches for peptides with two or more mutations becomes very high due to the Bonferroni correction to account for a huge size of the resulting (virtual) database that includes all mutated peptides [51].

To solve the MutMBPM problem, we will modify the pseudocode of `MBPMopt` in Figure 3.7. Given a pattern $P = p_1, \ldots, p_n$, we define its (i, δ, j, a)-*extension* as pattern $p_1, \ldots, p_{i-1}, p_i + \ldots + p_{j-1} + a + p_{j+1} + \ldots + p_{i+\delta-1}, p_{i+\delta}, \ldots, p_n$, where $1 \leq \delta \leq n - i + 1$, $i \leq j < i + \delta$, $a \in \Sigma$ and $a \neq p_j$. We define $P(i, d, \mathcal{P})^+$ of P as the set of d-bounded patterns resulting from all (i, δ, j, a)-*extension*'s for all $i \leq j < i + \delta$, and all $a \in \Sigma$ such that $a \neq p_j$. Additionally, each pattern P' in the set $P(i, d, \mathcal{P})^+$ must be $(i, \mathcal{P})$-compliant.[4] To accommodate for insertions, we relax the definition of a to $p_j + b$, where $b \in \Sigma$, and for deletions a can be 0. Although this definition is not included in the previous description of $P(i, d, \mathcal{P})^+$, it can be easily adapted. With these definitions in place, Figure 3.9 describes the pseudocode for the algorithm `MBPMmut` that solves the MutMBPM problem.

When the algorithm described in Figure 3.9 is implemented, $\mathcal{T}$ can be very large when i is very small (same size as $\mathcal{P}$ when $i = 1$). This can be a performance bottleneck in practice because $\mathcal{T}$ needs to be created for each k-mer. Therefore, we only substitute P by $P(i, d, \mathcal{P})^+$ when i reaches a threshold. In practice, the threshold is set to be 3 because gapped peptides have a minimum size of 6. This significantly reduces the size of $\mathcal{T}$ when the algorithm is run, but no matches with mutations in the first half of the patterns are retrieved. To solve this problem, we perform the search at a second pass, but in the reverse order. We evaluate the patterns in $\mathcal{P}$ in reverse, matching the last item in the patterns first. In Figure 3.9, the code can be easily adapted if all the references to prefixes are substituted by suffixes and T is reversed. The result is that the first pass will retrieve matches with no mutations in the first half of the patterns and the second pass will complement the first pass with matches with no mutations in the second half of the pattern.

[4] Although the set of all (i, δ, j, a)-*extension*'s could be potentially large because we have to try all possible values of a, the actual number of valid values of a resulting in a nonempty set of patterns that are $(i, \mathcal{P})$-compliant is much smaller. The implementation optimizes the algorithm so that all values of a are not evaluated.

`MBPMmut`$(T, \mathcal{P}, d)$

1: **for** $l = 1 \ldots |T|$ **do**
2: $\quad \mathcal{T} \leftarrow k$-mer at T_l
3: $\quad i \leftarrow 1$
4: $\quad$ **while** $\mathcal{T}$ is not empty **do**
5: $\quad\quad$ **for all** $P \in \mathcal{T}$ **do**
6: $\quad\quad\quad$ **if** i-prefix of P forms a block starting at T_l **then**
7: $\quad\quad\quad\quad$ substitute P by $P(i, d, \mathcal{P})^+ \cup P(i, d, \mathcal{P})$
8: $\quad\quad\quad$ **else**
9: $\quad\quad\quad\quad$ substitute P by $P(i, d, \mathcal{P})$
10: $\quad\quad\quad\quad$ **if** i-prefix of P matches $Q \in \mathcal{P}$ **then**
11: $\quad\quad\quad\quad\quad$ pattern Q matches text T with one mutation
12: $\quad\quad\quad\quad$ **end if**
13: $\quad\quad\quad$ **end if**
14: $\quad\quad$ **end for**
15: $\quad\quad i \leftarrow i + 1$
16: $\quad$ **end while**
17: **end for**

Figure 3.9: Mutated MBPM algorithm for matching a set of d-bounded patterns $\mathcal{P}$ against the text T by partitioning T into individual k-mers.

3.2.4 Modification-Tolerant Peptide Identification

The Modified Multiple Blocked Pattern Matching (ModMBPM) Problem models identification of peptides with post-translational modifications. These modifications are characterized by *offsets*, e.g., offset $\delta = 16$ Da characterizes oxidation of methionine. Let $T_i(\delta)$ be a text obtained from a text $T = t_1, \ldots t_n$ by adding δ to the i-th symbol of text T (for $1 \leq i \leq n$). A *modified block* in the text T is a block in $T_i(\delta)$ (for some i and δ). For example, if $T = 57, 112, 113, 113, 186$, $T_3(16) = 57, 112, 129, 113, 186$. A *modified block* in the text T is a block in $T_i(\delta)$ (for some i and δ). We are interested in the following problem:

Modified Multiple Blocked Pattern Matching (ModMBPM) Problem

Input. A text T over Σ and a set $\mathcal{P}$ of patterns.

Output. All modified blocks B in the text T such that $\overline{B} = P$, for some $P \in \mathcal{P}$.

Given a pattern $P = p_1, \ldots, p_n$, we define its (i, δ, j, m)-*extension* as pattern $p_1, \ldots, p_{i-1}, p_i + \ldots + p_j + m + p_{j+1} + \ldots + p_{i+\delta-1}, p_{i+\delta}, \ldots, p_n$, where $1 \leq \delta \leq n - i + 1$, $i \leq j < i + \delta$ and $-50 \leq m \leq 200$, indicating the mass range of the modification. We define $P(i, d, \mathcal{P})^*$ of P as the set of d-bounded patterns resulting from all (i, δ, j, m)-*extension*'s for all $i \leq j < i + \delta$, and $-50 \leq m \leq 200$. Additionally, each pattern P' in the set $P(i, d, \mathcal{P})^*$ must be $(i, \mathcal{P})$-compliant.[5] To solve ModMBPM problem, we use the set $P(i, d, \mathcal{P})^*$ instead of the set $P(i, d, \mathcal{P})^+$ at line 7 of the pseudocode in Figure 3.9. Line 11 of the algorithm should also be updated to reflect that the algorithm is matching modified blocks.

3.3 Results

We implemented the various algorithms described in the paper, but we only present results for the `MBPMopt`, `MBPMmut` and `MBPMmod` algorithms.

[5]The set $P(i, d, \mathcal{P})^*$ seems very large and its construction seems very inefficient because m can take on 251 values for each different value of the other parameters. However, in the implementation, valid values of m that result in $(i, \mathcal{P})$-compliant patterns are unique for each δ. Furthermore, for modifications, the values of j and m are independent.

3.3.1 Datasets

For benchmarking the speed of our algorithms, we used spectra from the organism *Shewanella oneidensis*, collected with LTQ-FT instruments from Richard Smith's laboratory at PNNL. The dataset has been extensively studied previously by Gupta et al. 2007 [45]. We arbitrarily chose over 200,000 spectra for benchmarking because the keyword tree of the gapped peptides from such dataset fits in main memory maximizing the performance of the search. For mutation search, we used over 200,000 spectra from *Arthrobacter* sp. strain FB24 generated using LTQ-FT instruments at PNNL. The organism was arbitrarily chosen to study alternative start codons in prokaryotes. Finally, for the modification search, we used the trypsin dataset analyzed by Kim et al. 2010 [52]. The dataset consists of over 170,000 spectra generated using collisionally induced dissociation (CID) and electron-transfer dissociation (ETD) fragmentation methods (equal number of spectra for each method) from human cell lysate. The detailed description of the dataset can be found in the original study [52].

3.3.2 Benchmarking

We benchmarked `MBPMopt` against InsPecT [3], one of the fastest algorithms for standard MS/MS database searches for unmodified peptides. We further benchmarked `MBPMmut` and `MBPMmod` against MS-Alignment [32] (also implemented in InsPect). `MBPMmut` and `MBPMmod` applied to gapped peptides are equivalent to MS/MS database searches for mutated peptides or peptides with unexpected modifications, respectively. `MBPMopt` results in a peptide identification tool that is an order of magnitude faster than InsPecT, while `MBPMmut` and `MBPMmod` are orders of magnitude faster than MS-Alignment (for large protein databases). To evaluate the speed of various peptide identification tools, we measure the time to match a million spectra against a proteome consisting of a million amino acids (this metric is measured in *secs per mil* [2]). It is estimated that the speed of Sequest [1] and InsPecT [3] is $\approx 1.8 \times 10^5$ and $\approx 3 \times 10^3$ secs per mil^2 correspondingly (see [10, 8]). MS-Alignment was run on a small sample of 1,000 spectra from the *Shewanella oneidensis* (see below). It took 1,140 seconds for MS-Alignment to complete the

run agains the proteome (1,375,623 amino acids), resulting in a speed of 8.3×10^5 secs per mil^2. Due to time constraints, we did not run the test for larger datasets, but the projected results are accurate as the algorithm scales linearly against larger datasets.

Our software tool consists of a de novo generation of gapped peptides by MS-GappedDictionary, followed by a pattern matching stage by one of the algorithms to solve the MBPM problem. The running time can be divided into preprocessing spectra (which includes generating the gapped peptides by MS-GappedDictionary) and matching gapped peptides against the database. The preprocessing stage is independent of the size of the database and the time is proportional only to the size of the spectral dataset. In practice, it takes on average 0.2 seconds to process a spectrum.

The running time of the second stage depends linearly on the size of the database. `MBPMopt` has a speed of 308 sec per mil^2. The speed was calculated based on a run of the tool on the *Shewanella oneidensis* dataset. It took 89 seconds for `MBPMopt` to complete a search of 210,192 spectra[6] on the proteome of *Shewanella oneidensis* consisting of 1,375,623 amino acids. The run was done on a modern desktop computer (Intel Core i7-965, 3.20 Ghz with 24GB of RAM).

`MBPMmut` and `MBPMmod` were run on the same *Shewanella oneidensis* dataset, For `MBPMmut`, it took 1,247 seconds to match the same 210,192 spectra to the *Shewanella oneidensis* proteome, yielding a speed of 4.3×10^3 secs per mil^2. For `MBPMmod`, it took 998 seconds to match the same dataset to the *Shewanella oneidensis* proteome, yielding a speed of 3.5×10^3 secs per mil^2. Figure 3.10 shows the plot that compares the projected times for our algorithms against InsPecT and MS-Alignment.

3.3.3 Gene Annotations in *Arthrobacter*

In this section we use MBPMmut to analyze a dataset of 221,673 spectra from *Arthrobacter* sp. strain FB24 generated in Richard Smith's laboratory at

[6]MS-GappedDictionary generated 10,724,012 gapped peptides from these spectra (51 gapped reconstructions per spectrum on average).

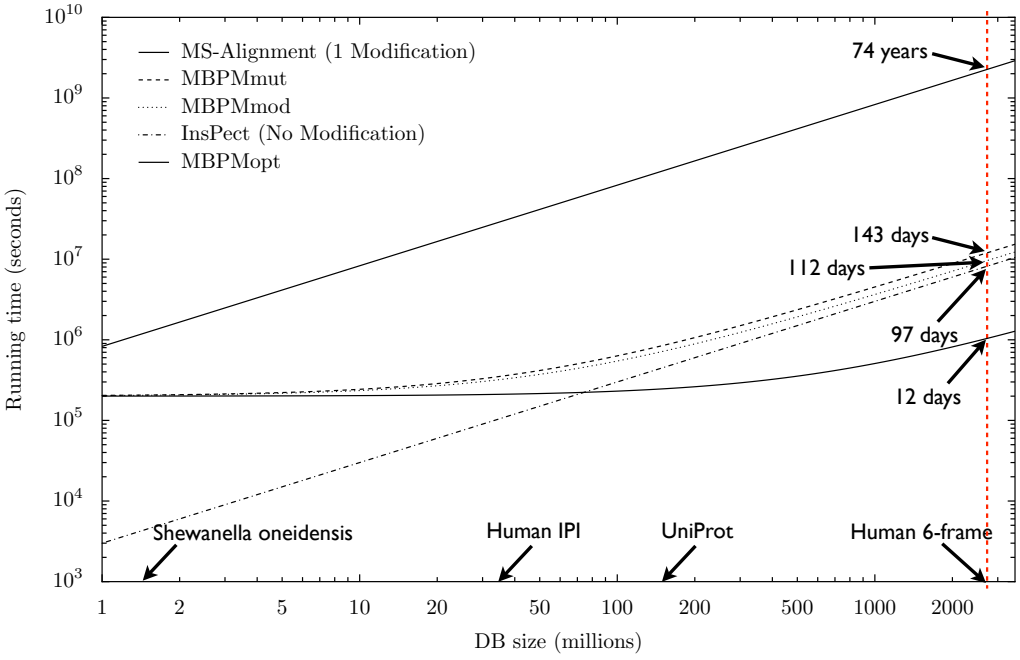


Figure 3.10: Performance of MBPM algorithms against InsPect. `MBPMopt` outperforms InsPect as soon as the database size exceeds 75 million amino acids. For the six-frame translation proteome of human (2.8 billion amino acids), `MBPMopt` is around 8 times faster than InsPect. `MBPMmod`, which searches for unexpected modifications in the database is only 15% slower than InsPect for regular database searches with no modifications.

PNNL using a LTQ-FT tandem mass spectrometer. 68,580 non-modified spectra were identified in search against the six-frame translation of the *Arthrobacter* genome of $\approx$ 10.1 Mb amino acids with a spectral probability threshold of 1.7^{-10} [41] at a 1% FDR. We analyzed the remaining 153,093 spectra using the MBPMmut algorithm. We transformed spectra into gapped peptides, searched them against the six-frame translation of *Arthrobacter* genome using a probability cut off of 1.5e-15 to achieve a 1% FDR.[7]

To identify alternative start codons, we collected all identified peptides with a mutation in the 1st position and with spectral probability better than 1.5e-15. Out of 189 such peptides, the most prevalent mutations can be alternatively explained as precursor mass errors (1 or 2 Da offsets), acetylation (42 Da offset) and oxidation (16 Da offset). For example, a Glu to Gln mutation ($\approx$ 1 Da offset) can be explained by a precursor mass error, while a Ser to Glu mutation ($\approx$ 42 Da offset) can be explained as acetylation. These alternative explanations represent useful peptide identifications (but not mutations) and account for $\approx$ 70% of the identified peptides.

The next class of the most prevalent mutations represent amino acids that mutate into methionine and reveal potential alternative start codons. The most common mutations among these peptides were valine into methionine (4 peptides and 37 spectra) and leucine into methionine (3 peptides and 17 spectra) in the first position. All mutations from valine and leucine to methionine represented GTG and TTG, potential start codons (See Table 3.1). All seven predictions were verified to be start codons by mapping back the identified peptides to the annotated proteome of the organism.

3.3.4 Blind modification search

In this section, we describe the results of the `MBPMmod` algorithm on the human CID/ETD trypsin dataset. The dataset contains 179,440 spectra. We identified 58,419 spectra using the MBPMopt algorithm and matching them against

[7]The goal of this section is to illustrate the capabilities of the `MBPMmut` and `MBPMmod` rather than to provide a comprehensive re-annotation of *Arthrobacter*.

Table 3.1: Peptide matches with mutations resulting in Methionine at the first position. The first amino acid to the right of the colon represents the amino acid derived from the genome using the standard translation table. The codon column represents the original codon in the six-frame translated proteome. The second column represents the best MSGF spectral probability [41] among all the identified spectra for the given spectra.

Identification	Best Prob	Codon	# Spec
(V:M)NTPTSVTASPPDLAGEEPK	3.4e-18	GTG	4
(V:M)LIAQRPTLSEEVVSENR	9.4e-19	GTG	13
(V:M)IEETLLEAGDKMDK	2.4e-19	GTG	14
(V:M)STVESLVGEWLPLPDVAEMMNVSITK	2.7e-18	GTG	6
(L:M)LTANAYAAPSADGDLVPTTIER	6.6e-18	TTG	3
(L:M)EGPEIQFSEAVIDNGR	2.6e-18	TTG	12
(L:M)DTTVADTEVTMPEGQGPR	1.4e-21	TTG	2

the human IPI database version 3.78. The identified spectra passed the 1% FDR with spectra probability better than 4.0e-11. Table 3.2 summarizes the modifications that are represented by more than 20 sites. We identified an additional 9,387 modified spectra using a 1% FDR with spectra probability better than 1.3e-15.

3.4 Discussion

We introduced a new class of combinatorial pattern matching problems and proposed various algorithms for their solution. These algorithms represent the first practical applications of BPM in proteomics. Our results demonstrate that BPM has a potential to greatly speed up protein identifications, a key task in computational proteomics.

3.4.1 Fused Blocked Pattern Matching Problem.

In this subsection, we present the possibility of using the MBPM algorithm to find fused peptides, and describe the shortcomings when the approach is applied to gapped peptides generated by MSGappedDictionary. Given a text $T = t_1, \ldots t_n$, we define a text $T_{i,j} = t_1, \ldots t_{i-1}, t_i, t_j, t_{j+1}, \ldots, t_n$ where i and j are not consecutive. A *fused block* is a block in $T_{i,j}$ (for $1 \leq i, j \leq n$) that is not a block in T. For example, substrings (57,112) and (113,186) form a fused block in

Table 3.2: Observed modifications in the human lysate dataset. The mass shift represents the average observed mass shift of the identifications and due to mass errors, it might not correspond to the exact mass of the modification listed in UNIMOD. We discarded modifications of 1 or 2 Da because they are the result of parent mass errors rather than meaningful modifications.

Name	Mass	Spectra	Sites	Unimod ID [53]
Dethiomethyl	-47.994	97	64	526
Loss of ammonia/Pyro-glu from Q	-17.018	248	103	385/28
Methylation	14.021	46	20	34
Oxidation or Hydroxylation	16.002	542	236	35
Dimethyl	28.019	39	25	36
Acetylation	42.020	156	53	1
Carbamyl	43.013	296	128	5
Carboxylation	44.008	34	22	299
Carbamidomethyl	57.029	535	261	4
Carboxymethyl	58.020	63	34	6
Phosphorylation	79.967	90	61	21

the text $T = 57, 112, 113, 113, 186$ because they form a block in $T(2, 4)$. Formally, the problem can be described:

Multiple Fused Blocked Pattern Matching Problem

Input. A text T over Σ and a set $\mathcal{P}$ of patterns.

Output. All fused blocks B in the text T such that $\overline{B} = P$, for some $P \in \mathcal{P}$.

The approach to solving the Fused Pattern Matching Problems is based on the observation that for every pattern $p_1, \ldots, p_n$ matching a fused block in the text, its prefix $p_1, \ldots, p_m$ and its suffix $p_{m+1}, \ldots, p_n$ matches a (non-mutated) block in the text. It is trivial to see that all partial prefix matches can be retrieved using the algorithm in Figure 3.7 by storing all i-prefixes that match a prefix in $\mathcal{P}$ (lines 7 and 8). To retrieve all partial suffix matches, we can also use the algorithm in Figure 3.7, but we need to reverse the k-mer and substitute all references of prefixes with suffixes (in addition to changes to line 7 and 8). After all the partial prefixes and suffixes are stored, we find all blocks $B_{prefix}(i)$ matching the prefix $p_1, \ldots, p_{i-1}$ and all blocks $B_{suffix}(i)$ matching the suffix $p_{i+1}, \ldots, p_n$, correspondingly (for all $1 < i < n$) and for all patterns $P = p_1, \ldots, p_{i-1} \in \mathcal{P}$. Notice that retrieving all fused blocks, $B_{prefix}(i)$ and $B_{suffix}(i)$, takes time proportional to the length

of the pattern (plus the time for constructing all results) because all the partial matches are pre-computed during the matching stage. We remark that in practice, the identification of fused peptides is typically limited to long peptides with the fusion break located closely to the middle of the peptide [42]. In such cases, both $p_1, \ldots, p_i$ and $p_{i+1}, \ldots, p_n$ are rather long and thus the sets $B_{prefix}(i)$ and $B_{suffix}(i)$ are expected to be small.

One additional detail is that the fusion point is not required to coincide with the boundary of p_i and $p_i + 1$. Therefore, we also allow reconstructions of $B_{prefix}(i-1)$ and $B_{suffix}(i)$, where p_i partially matches the prefix and the suffix text.

However, this algorithm does not scale well with large proteomes. One of the requirements for reliable identifications is that the peptide, and consequently the spectrum, must be large. The gapped peptide reconstructions from these peptides typically have very large gaps at the prefix or suffix ends (greater than 2000 daltons). As a result, the fusion point is covered by the large gap at one of the ends making either the set $B_{prefix}(i)$ or $B_{suffix}(i)$ unmanageably large. The Fused BPM algorithm will only be practical for gapped peptides that are more symmetric (e. g. all gaps are no greater than 300 Da), but for the current implementation of MSGappedDictionary, this algorithm is not practical.

3.5 Acknowledgements

This chapter is in preparation for publication as "Blocked Pattern Matching Problem and its Applications in Computational Proteomics". J. Ng, and P. A. Pevzner 2011, in preparation. The dissertation author is the primary author of this paper.

Chapter 4

Dereplication and De Novo Sequencing of Nonribosomal Peptides

Nonribosomal peptides (NRPs) are of great pharmacological importance, but there is currently no technology for high-throughput NRP *dereplication* and sequencing. We used multistage mass spectrometry followed by spectral alignment algorithms for sequencing of cyclic NRPs. We also developed an algorithm for comparative NRP dereplication that establishes similarities between newly isolated and previously identified similar but nonidentical NRPs, substantially reducing dereplication efforts.

4.1 Introduction

The classical protein synthesis pathway (translation of template mRNA) is not the only mechanism for cells to assemble amino acids into proteins or peptides. Nonribosomal peptide synthesis is performed by nonribosomal peptide (NRP) Synthetases that represent both the mRNA-free template and building machinery for the peptide biosynthesis [54]. NRP synthetases produce NRPs that are not directly inscribed in genomes and thus cannot be inferred with traditional DNA sequencing. NRPs are of great pharmacological importance as they have been

optimized by evolution for chemical defense and communication. Starting from penicillin, NRPs and other natural products have an unparalleled track record in pharmacology: most anticancer and antimicrobial agents are natural products or their derivatives [55]. NRPs include antibiotics, antiviral and antitumor agents, immunosuppressors and toxins.

Most NRPs contain nonstandard amino acids, increasing the number of possible building blocks from 20 (in standard ribosomal peptides) to several hundred (See Table A.1). Previous methods for NRP characterization are based on nuclear magnetic resonance (NMR) spectroscopy and are time-consuming and error-prone [56, 57, 58, 59]. Therefore, there is a need for the efficient structure elucidation of NRPs. Furthermore, substantial efforts in activity screening can be saved if newly isolated compounds can be rapidly associated to a known compound by *dereplication* [60]. Dereplication refers to the process of screening for active compounds in a mixture discarding those that have been previously studied to avoid recharacterization.

In a pioneering study [61], a cyclic algal peptide had been linearized and manually sequenced using tandem mass spectrometry (MS^2). This approach, although successful, did not result in a robust NRP sequencing technique as most NRPs evade linearization attempts. Characterization of hormothamnin A is another example of mass spectrometrybased NRP sequencing [62]. Furthermore, structural variants of antimicrobial agent tyrothricin had been characterized from a mixture of NRPs [63], using tandem mass spectrometry. In a similar experiment [64], new variations of streptocidins had also been sequenced. However, no automatic tool had been created from these studies.

We compared spectra of similar but nonidentical NRPs, enabling *comparative dereplication* that establishes the similarity between a newly isolated and a previously identified similar (rather than identical) compounds. This is in contrast to the classical definition of dereplication, which only considers identical compounds. Because many NRPs are produced as related analogs (for example, 61 out of 90 cyanopeptides recently identified in drinking water are variants of known peptides [65]), comparative dereplication can reduce NRP characterization

efforts from weeks to minutes. For example, cyanopeptide X was an unknown bioactive compound (currently known as desmethoxymajusculamide C) when our project started in 2007, but was sequenced using NMR spectroscopy in 2008. The effort invested in analyzing this NRP in 2007 would have been saved if our algorithm, NRP-dereplication, were available. Indeed, NRP-dereplication revealed that cyanopeptide X is related to majusculamide C. Another example is compound 879 that had been assumed to be new but was found to be known during the patent application. NRP-dereplication revealed that compound 879 is neoviridogrisen. NRP-dereplication derives a sequence of an unknown compound given a database of known cyclic peptides (provided a related peptide is known). In the cases when no related NRPs are known, we performed de novo sequencing with NRP-sequencing, a self-alignment-based algorithm, and NRP-tagging, an approach that uses frequently occurring amino acid tags for peptide reconstruction. We also reconstructed cyanopeptide X, which is to our knowledge the first report of automated de novo reconstruction of a cyclic peptide by mass spectrometry.

4.2 Methods

4.2.1 Data acquisition and preprocessing

Seglitide, tyrocidines, BQ-123, destruxin A and microcystin LR were purchased from Sigma-Aldrich. H-3526 and H-8405 were purchased from Bachem. Cyanopeptide X, cyclomarins and compound 879 were provided by Gerwick's, Moore's and Fenical's laboratories at University of California, San Diego, respectively.

Time-of-flight (TOF) mass spectrometry data was acquired for tyrocidine A, A1, B, B1, C, C1; cyclomarin A, C; dehydrocyclomarin A, C; BQ123; microcystin LR; compound 879; and H8405. Ion-trap mass spectrometry data were acquired for seglitide, cyanopeptide X, destruxin A and H3526.

For the ion-trap data acquisition, each compound was prepared to 1 μM solution using 50:50 MeOH:water with 1% AcOH as solvent, and underwent nanoelectrospray ionization on a Biversa Nanomate (pressure: 0.3 p.s.i., spray voltage:

1.4-1.8 kV). Ion trap spectra were acquired on a Finnigan LTQ-MS (Thermo-Electron Corporation) running Tune Plus software version 1.0. For the MS^n data collection, spectrum ion trees were collected in both automatic mode and manual mode. In automatic mode, the [M+H]+ of each compound was set as the parent ion. MS^n data were collected with the following parameters: maximum breadth, 20; maximum MS^n depth, 3. At $n = 2$, isolation width, 4; normalized energy, 50. At $n = 3$, isolation width, 4; normalized energy 30. For manually collected data, the [M+H]+ ion of each compound was isolated with an isolation width of 3 mass to charge (m/z) units and fragmented with normalized collision energy of 30. Top 20 intense ions within the spectra were isolated with an isolation width of 3 m/z units and fragmented with normalized collision energy of 30. The Thermo-Finnigan files (in RAW format) were then converted to an mzXML file format using the ReAdW (http://tools.proteomecenter.org/).

For the TOF data collection, the cyclic peptides were prepared in a 50% methanol, 0.5% AcOH at 1 pmol/μl. The samples were then infused into an ABI QSTAR XL QTOF using nanospray source I for ionization at 0.5 l/min. The instrument was then set up in automatic acquisition mode to collect one MS scan to detect the calibrants (CsCl (Sigma) and cPDI inhibitor (Bachem)) and one product ion scan for the parent mass of the peptide in the experiment. Each scan time was 30 s and the method length was 2 min. The acquisition was set to enhance for the scanned ranges. The collision energy for each compound was determined using direct infusion in tune mode to find out the optimal collision energy required to produce ideal fragmentation for MS^2. The collected spectra were calibrated using the first mass spectrometry scan and the calibration was applied to the entire file.

All spectra were preprocessed before the sequencing algorithms were applied. The initial filtering steps were to ensure that the low-intensity peaks are removed. The standard procedure of keeping the top 5 peaks within a window of 50 Da was applied to all compounds.

4.2.2 Mass Spectra of a Cyclic Peptide

When analyzing a cyclic peptide using mass spectrometry, the MS^2 stage amounts to breaking (linearizing) the cyclic peptide into linear peptides with the same parent mass (Figure 4.1A-E). The next stage of mass spectrometry (MS^3) breaks the different linearized versions of the cyclic peptide, resulting in the difficult problem of interpreting a MS^3 spectrum of different (but related) peptides. The theoretical MS^3 spectrum, $Spectrum(P)$ of the cyclic peptide $P = p_1 \dots p_n$ is thus the superposition of the theoretical spectra, $Spectrum(P_i)$ of n linear peptides $P_i = p_i \dots p_n p_1 \dots p_{i-1}$ for $i = 1 \dots n$ (Figures 4.1 and 4.2).

4.2.3 Comparative Dereplication

Comparative dereplication can be formulated as the *Cyclic Peptide Dereplication Problem (CPDP)*: Given an experimental spectrum S, a cyclic peptide P, and a parameter k (maximum number of mutations/modifications), find a cyclic peptide P' with at most k mutations/modifications from P that maximizes the number of shared masses between S and the theoretical spectrum of P'.

We address the CPDP problem for the most relevant case $k \leq 1$. Given the MS^3 spectrum of an unknown peptide P', and the sequence of a known peptide P that differs from P' by a single mutation at an (unknown) position x, NRP-Dereplication derives P'. NRP-Dereplication is based on the observation that most peaks shared between the experimental spectrum of P' and theoretical spectrum P correspond to subpeptides that do not contain position x (*0-correlated* subpeptides). Conversely, most peaks in the experimental spectrum P' that differ from the peaks in the theoretical spectrum of P by $\delta = Mass(P') - Mass(P)$ correspond to subpeptides that contain position x (δ-*correlated* subpeptides). The *coverage* of a position x is defined as the number of 0-correlated subpeptides containing that position, plus the number of δ-correlated subpeptides not containing that position. Thus, *correlated* subpeptides (both 0-correlated and δ-correlated) have a potential to reveal the differing amino acid as the amino acid with the minimum coverage. For example, the drop in coverage at ornithine (Figure 4.3) allows one to dereplicate the experimental spectrum of tyrocidine C1 using sequence of

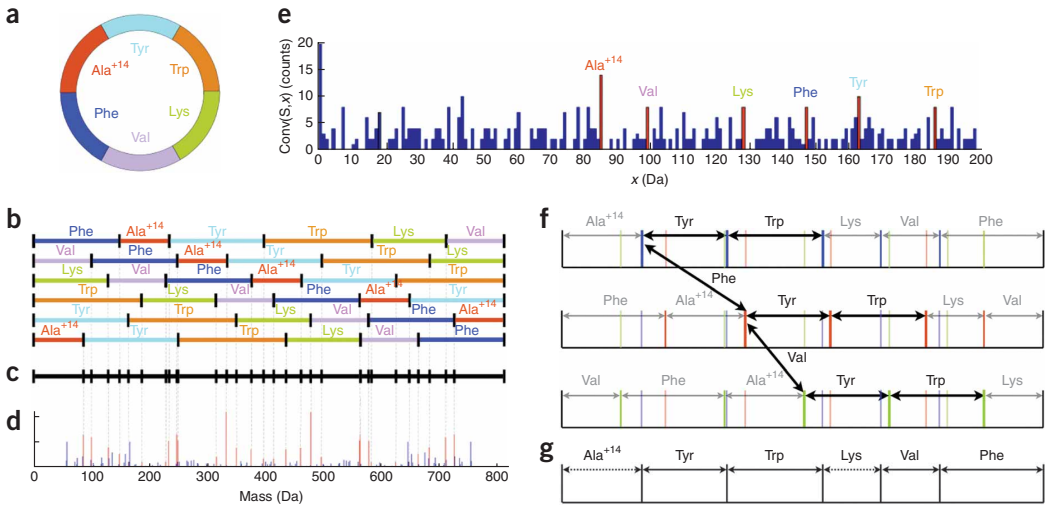


Figure 4.1: Experimental and theoretical spectrum of seglitide, *Cyclic*(N-methyl-Ala–Tyr-D–Trp–Lys–Val–Phe), a somatostatin receptor antagonist. A) Cyclic diagram of seglitide. Ala^{+14} represents methylated alanine. The integer residue masses are 85, 163, 186, 128, 99 and 147 Da corresponding to cyclic A^{+14}YWKVF. B) Representation of the six different theoretical linear peptides after MS^2 fragmentation of seglitide (cyclic). C) Superposition of the theoretical linear fragments from (B). D) Experimental spectrum of seglitide (the peaks corresponding to fragment masses in the theoretical spectrum of segtilite in (C) are shown in red). E) Autoconvolution of the spectrum in insert (D) showing prominent peaks for offsets corresponding to masses of amino acids (shown in red). The peak at 0 is truncated. F) Three identical theoretical spectra of seglitide annotated as A^{+14}YWKVF (blue), FA^{+14}YWKV (red) and VF^{A+14}YWK (green) illustrating the occurrences of amino acid tags. The frequent 2-amino-acid tag Tyr-Trp was observed in three different locations in the spectrum. Additionally, the offsets between three consecutive locations of tag Tyr-Trp revealed the masses of amino acids phenylalanine and valine. G) The gapped peptide constructed from f combines Tyr-Trp (derived from a frequent tag) with Val-Phe (derived from the inter distances between tag locations). Ala^{+14} and Lys were inferred from the flanking masses of Tyr-Trp and Val-Phe. The complete sequence A^{+14}YWKVF was recovered for seglitide, but gaps may be generated for larger compounds.

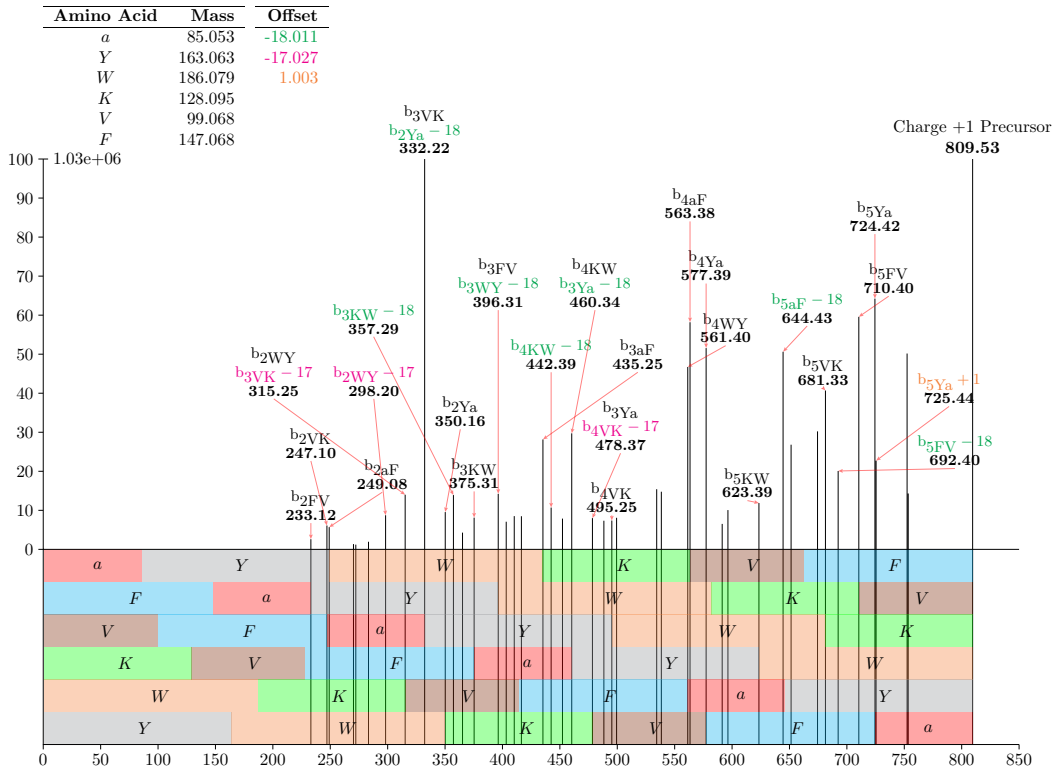


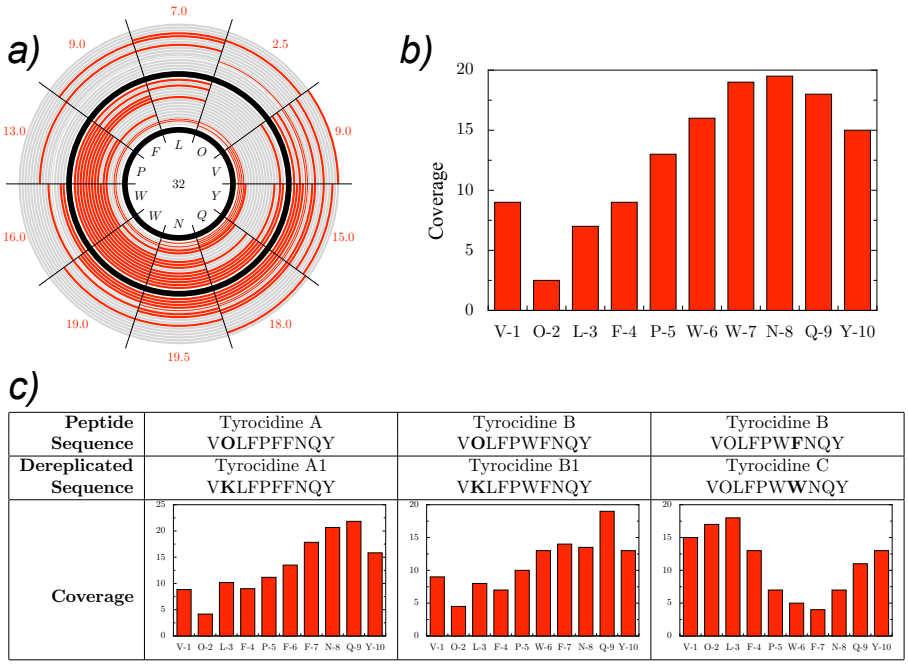
Figure 4.2: Annotation of the experimental spectrum of seglitide. The x-axis represents m/z values, while the y-axis represents relative intensity. The number to the right of the y-axis (1.03e+6) represents the absolute intensity at 100%. The Offset column lists the neutral losses considered when annotating the spectrum.

tyrocidine C.

As the peptide P to be used for dereplication is not known in advance, every NRP spectrum needs to be compared to a database of known cyclic peptides such as Norine [66]. NRP-Dereplication can localize the single mutation using the top-scoring peptide in the Norine database (See Table 4.2).

The tyrocidine family presents an ideal test for NRP-Dereplication because tyrocidine A, B and C are in the Norine database, whereas tyrocidines A1, B1 and C1 are not. NRP-dereplication showed that spectra from tyrocidine A, B and C had top hits corresponding to Norine-database peptides, whereas their A1, B1 and C1 counterparts were mapped to high-scoring matches with one mutation (Table 4.2). The correct mutated position is also localized by NRP-Dereplication as the position with minimum coverage for all compounds we analyzed that had a closely related compound in the NRP database. NRP-Dereplication generated only two high-scoring false hits representing very short peptides (H8495 and BQ123), but closer examination revealed that the matches were correlated to the query peptides. We conducted additional experiments that demonstrated that NRP-Dereplication can localize the correct position of the mutation when $k = 1$ (Figure 4.4).

While Figure 4.3 demonstrates that drops in coverage reveal the differing amino acid, we need to ensure that "random" pairs of peptides do not exhibit similar drops (otherwise dereplication will fail when comparing a spectrum against a database of known NRPs). The number of correlated subpeptides for random peptides pairs is much smaller than for related peptide pairs. In each dereplication experiment, we compared the tyrocidine C experimental spectrum to an (incorrect) peptide that differed from the correct peptide by a fixed number of amino acids. While there are 32 correlated subpeptides between the experimental tyrocidine C1 spectrum and the tyrocidine C peptide (differing by a single amino acid), the number of these subpeptides quickly decreases as the peptides diverge (Figure 4.4A). Our simulations revealed that NRP-Dereplication is correct in over 90% of cases in case of a single amino acid difference (Figure 4.4C). As expected, the number of correlated subpeptides drops as the number of differing amino acids increases (see the Average Worst Rank plot in Figure 4.4B).



Peptide Sequence	Tyrocidine A VOLFPFFNQY	Tyrocidine B VOLFPWFNQY	Tyrocidine B VOLFPW**F**NQY
Dereplicated Sequence	Tyrocidine A1 V**K**LFPFFNQY	Tyrocidine B1 V**K**LFPWFNQY	Tyrocidine C VOLFPW**W**NQY
Coverage			

Figure 4.3: Dereplication results of tyrocidines. A) NRP-Dereplication output for experimental spectrum of tyrocidine C1 (V**K**LFPWWNQY) given peptide sequence of tyrocidine C (V**O**LFPWWNQY). Concentric red-gray circles represent 0-correlated subpeptides (with peptide shown red and its complement shown gray) and δ-correlated subpeptides (with peptide shown gray and its complement shown red). Given this coloring convention, the amino acid coverage (number of red arcs covering an amino acid) represents supporting evidence that an amino acids did *not* change from the known to the unknown compound. The thick black circle separates 0-correlated subpeptides (shown inside) from δ-correlated subpeptides (shown outside). The outer counts represent the coverage for a given amino acid by red arcs and reveals the differing amino acid (O) as the amino acid with minimum coverage (2.5 vs. 7 for the next runner-up). The counts are normalized by the number of subpeptides per peak. For example, if a peak has two alternative subpeptide annotations, it will contribute $\frac{1}{2}$ to the coverage. The width of the arcs are proportional to this weighting factor. The number in the center of the graph is the total number of correlated subpeptides. B) Alternative representation of (A) as a histogram that reveals the changed amino acid O. C) Additional dereplication results for the tyrocidine family.

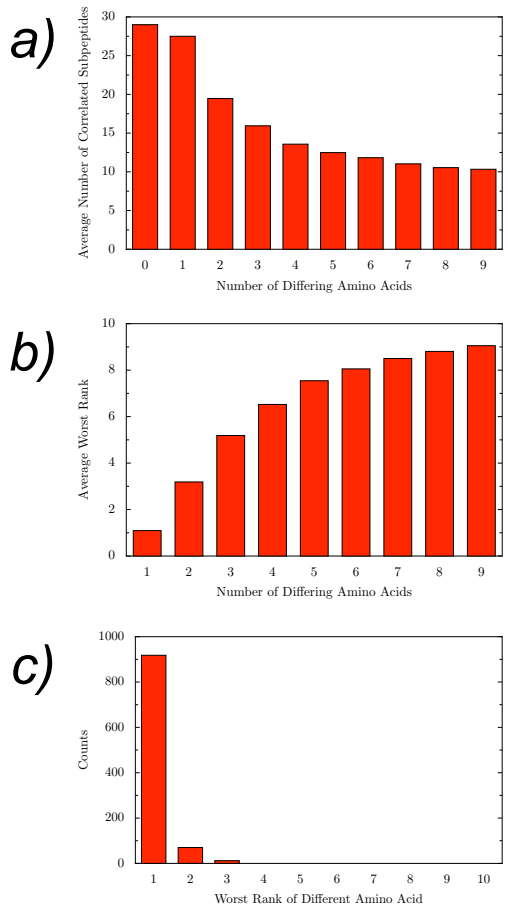


Figure 4.4: Dereplication results for the experimental spectrum of tyrocidine C (VOLFPWWNQY) compared against peptides with varying number of differing amino acids. All experiments were ran 1000 times randomly choosing a different offset(s) on a random amino acid(s). Each of the differing masses have a randomly chosen offset in the range of 15 to 43 Daltons (the sign of offset is also chosen randomly). A) The number of correlated subpeptides as a function of the number of differing amino acids. B) The average worst rank of the known differing amino acid(s). The worst rank is the highest rank for all the differing amino acids when sorted in the increasing order of their coverage. For example, if amino acids A and B were modified, and after the dereplication experiment, their coverage ranks were 1 and 3, the worst rank would be 3. In the case of k differing amino acids, a rank of k means that the dereplication experiment was successful. C) For case of a single amino acid difference, NRP-Dereplication algorithm is correct in over 90% of cases. We note that for the cases in which the differing amino acid has rank 2, the position with the lowest coverage is usually a neighboring position.

4.2.4 De Novo Sequencing

In the case where no related peptide is known (and thus NRP-Dereplication is not applicable), we formulated the *Cyclic Peptide Sequencing Problem (CPSP)*: given an experimental spectrum S, find a cyclic peptide P maximizing the number of shared masses between S and the theoretical spectrum of P. Reconstructing the cyclic peptide P from its theoretical spectrum, $Spectrum(P)$, amounts to the cyclic version of the partial digest problem [67].

However, it is not clear how to extend the algorithms for the partial digest problem [67, 68] to a cyclic setup. Furthermore, reconstructing P from its experimental MS^3 spectrum S is a difficult problem because the contributions of different linear versions of P to the experimental spectrum are nonuniform. However, spectral convolution and spectral alignment [69] can reveal similarities between related spectra. Because an MS^3 spectrum of a cyclic peptide is a superposition of spectra of related linearized peptides, spectral autoconvolution and autoalignment reveal key features of the cyclic peptide.

Autoconvolution of a spectrum S with offset x is defined as the number of masses s in S such that $s - x$ is also a mass in S. We defined the *cyclic* autoconvolution, $conv(S, x)$, as the number of masses s in S such that either $(s - x)$ or $(s - x) + precursorMass(S)$ is also a mass in S. For example, high-scoring positions of the autoconvolution of seglitide revealed masses of amino acids of the NRP (Figure 4.1E). Furthermore, the largest peak $conv(S, 85) = 14$ corresponded to the mass of the methylated alanine (Ala^{+14}). The other five amino acids in seglitide are also represented by prominent peaks at positions 99, 128, 147, 163 and 186 with $conv(S, x) \geq 8$, corresponding to their integer masses in daltons. Spectral autoconvolution (Figure 4.1E) is a computational approach to derive residue masses of cyclic peptides.

Autoalignment of a spectrum S with offset x is defined as the set of peaks $S_x = \{s : s \in S \text{ and } (s - x) \in S\}$. Autoalignment can be viewed as a virtual spectrum with parent mass $precursorMass(S) - x$ (Figure 4.5). For seglitide, S_{85} ($x = 85$ maximizes $conv(S, x)$ for seglitide) corresponds to the alignment between A^{+14}YWKVF and YWKVFA^{+14}.

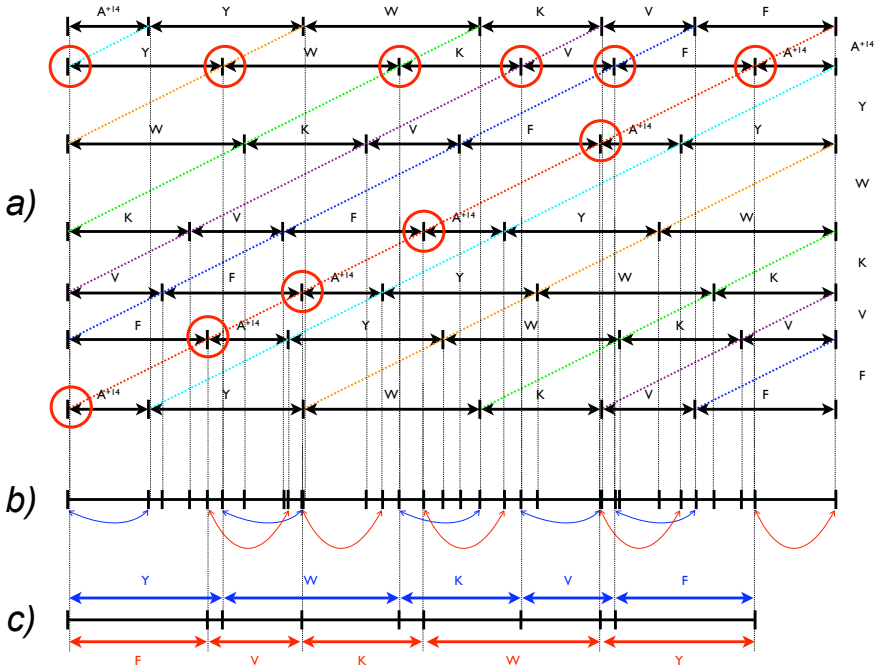


Figure 4.5: Autoalignment S_{85} of the theoretical spectrum of seglitide (85 Da represents the most prominent peak in auto-convolution). All peaks with mass s that have a related peak with mass $s+85$ are enclosed in red circles. A) Representation of the theoretical spectrum for seglitide. Each horizontal line represents a different linear version of seglitide (prefix ladder). The linearized version A^{+14}YWKVF is shown twice. The diagonal lines represent the suffix ladders. If we stay on a diagonal, and walk from left to right, we read out the mass sequence of seglitide in reverse. The vertical axis is drawn at half the scale of the horizontal axis to save space. B) Shows the cyclic theoretical spectrum of seglitide by compressing all breaks of (A) into a single line. There are two set of peaks that are 85 Da apart. First, those peaks in the first and second horizontal lines (aligned prefixes) and second, those peaks in the red and cyan diagonals (aligned suffixes). These sets are highlighted with the blue and red arcs in the cyclic theoretical spectrum, respectively. C) Autoalignment of (B) with offset of 85 Da. All matching peak pairs (identified by arcs) will be reflected in the consensus spectrum. The autoalignment spectrum contains the prefix and suffix ladder of the overlapping linearized seglitide sequences. Note that in reality, prefix and suffix ladders are not separated in the autoalignment spectrum.

Using the concepts of autoconvolution and autoalignment, we created NRP-Sequencing, an algorithm to solve the cyclic peptide sequencing problem that does not require prior knowledge of the amino acid masses in the compound. NRP-Sequencing first uses the MS^3 autoconvolution to derive the set of possible amino acid masses and then uses the MS^3 autoalignment using the top k possible offset masses, x, to construct a consensus spectrum S_x for each x. NRP-Sequencing then generates all possible reconstructions for each S_x and reranks all generated cyclic peptides according to their matches to the MS^n spectra (for $n = 3, 4$ and 5). Details on the NRP-Sequencing algorithm are as follow:

Input MS^3 spectrum S of an (unknown) cyclic peptide, set of MS^n spectra, parameter k (maximum number of candidate aa-masses) and p (minimum percentage of top de novo score to report a suboptimal peptide).

Output Ranked list of candidate peptide reconstructions

1: $PeptideList = \emptyset$
2: Select top k peaks $x_1 \dots x_k$ in the autoconvolution $conv(S, x)$ in the $[57, 200]$ Da interval
3: **for** $i = 1$ to k **do**
4: Set $x = x_i$ and construct the auto-alignment S_x
5: De novo sequence S_x and find highest scoring peptide P
6: For every suboptimal peptide P' such that $score(P') \geq p \cdot score(P)$
7: Append x to P' and add the resulting cyclic peptide to $PeptideList$
8: **end for**
9: Rescore each peptide P' in $PeptideList$ by matching P' against all MS^n spectra
10: Output peptides from $PeptideList$ in the decreasing order of their scores

In default mode, NRP-Sequencing selects the masses of the top 20 autoconvolution masses in the interval 57-200 Da and combines them with the masses of standard amino acids. NRP-Sequencing could generate the correct sequence (among the set of generated reconstructions) in all cases when the resulting set of masses contained all amino acid masses in the NRP (11 out of 18 compounds).

Moreover, in almost all cases the correct sequences were ranked as the top-scoring reconstruction (See Table 4.3). However, the success of NRP sequencing is constrained by the ability to determine all amino acid masses by autoconvolution.

Because some positions are less prone to breakage than others, recovering all amino acid masses in an NRP using autoconvolution may be an unattainable goal. NRP-Tagging attempts to reconstruct gapped peptides from MS^3 spectra of cyclic peptides (Figure 4.1G). Spectra of cyclic peptides are superpositions of related (cyclically shifted) linear peptides that tend to have the same tags repeated in the spectrum. Given an MS^3 spectrum, we found all 2-amino-acid tags XY (defined by triplets of peaks $s, s+X, s+X+Y$ in the spectrum) and selected all frequent tags (for example, tags repeated 3 or more times). For example, if a tag XY starts at positions $s, s+A$ and $s+A+B$, then masses A and B may represent two other (adjacent) amino acids in the cyclic peptide (Figure 4.1G). NRP-Tagging first constructs a gapped peptide (for example, 85, 163, 186, 128 and 246 Da for seglitide, indicating integer masses of single or combined amino acids) and then attempts to extend it into full-length de novo reconstructions (for example, 85, 163, 186, 128, 99 and 147 Da, indicating integer masses of amino acids in seglitide). As gapped peptides often contain masses representing combined masses of adjacent amino acids (for example, 246 = 99+147 Da), NRP-Tagging attempts to partition each mass in the gapped peptide into smaller masses. Details on the NRP-Tagging algorithm are as follow:

Input MS^3 spectrum S of an (unknown) cyclic peptide, a minimum tag *frequency*, a recursion *depth*, and a scoring function $score(S, peptide)$.

Output Ranked list of candidate gapped peptides.

1. Find all tags in S:
   1: **tags**$(x, y) = \{\}$ for all $0 \prec x, y \prec 200$
   2: **for all** $s, s', s'' \in S$ such that $s_i \prec s_j \prec s_k$ **do**
   3: $mass_1 = s' - s$
   4: $mass_2 = s'' - s'$
   5: add s to **tags**$(mass_1, mass_2)$

6: **end for**

2. Generate gapped peptides from frequent tags:

1: **gappedPeptides** = {}
2: **for all** $mass_1, mass_2$ with $|\textbf{tags}(mass_1, mass_2)| > \textit{frequency}$ **do**
3: **for all** $\{0 \prec s_1 \prec \ldots \prec s_n \prec mass(S) - mass_1 - mass_2\} \subseteq \textbf{tags}(mass_1, mass_2)$ **do**
4: *gappedPeptide* $= [m_1, ..., m_n, mass_1, mass_2, m_{n+1}]$ where $m_i = s_i - s_{i-1}$, for $2 \leq i \leq n$, $m_1 = s_1$ and $m_{n+1} = mass(S) - mass_1 - mass_2 - s_n$
5: Add *gappedPeptide* to **gappedPeptides**
6: **end for**
7: **end for**

3. Iteratively attempt to split masses larger than 200 Da:

1: **results** = *depth* top-scoring peptides from **gappedPeptides**
2: **candidates** = **results**
3: **repeat**
4: **sequences** = {}
5: **for all** *gappedPeptide* in **candidates do**
6: **intermediates** = {}
7: **for all** $mass > 200$ Da in *gappedPeptide* **do**
8: **for all** $mass_1$ such that $0 \prec mass_1 \prec 200$ Da **do**
9: split *mass* in *gappedPeptide* into $(mass_1, mass - mass_1)$ and add the resulting peptide to **intermediates**
10: **end for**
11: **end for**
12: add *depth* top-scoring peptides from **intermediates** to **sequences**
13: **end for**
14: **candidates** = **sequences**
15: Add **sequences** to **results**
16: **until sequences** is empty

17: **return results**

tags$(mass_1, mass_2)$ contains the starting positions of all tags formed by amino acids with masses $mass_1$ and $mass_2$. The notation $|\mathbf{tags}(mass_1, mass_2)|$ refers to the number of locations of a 2-amino-acid tag with masses $(mass_1, mass_2)$. The notation $x \prec y$ denotes that $y - x \geq 57$ (57 Da represents the mass of the smallest amino acid Gly). For a given set of starting positions in **tags**$(mass_1, mass_2)$, all possible combinations $(\{s_1 \prec \ldots \prec s_n\} \subseteq \mathbf{tags}(mass_1, mass_2))$ of starting positions of tags are considered during the gapped peptide reconstruction. The precursor mass of S is denoted as $mass(S)$. While the pseudocode above attempts to split each $mass > 200$ Da into all possible pairs $(mass_1, mass - mass_1$ with $0 \prec mass_1 \prec 200$, the real implementation only considers $mass_1$ as a splitting mass if it is supported by some peaks in S. There are 2 threshold parameters, *frequency* (minimum number of occurrences of a tag in S), and *depth* (limits the number of high scoring gapped peptides per an iteration of the mass splitting). The scoring function $score(S, peptide)$ is used to rank the intermediate peptides and select those for the next iteration.

Similar to algorithms for sequencing linear peptides, NRP-Tagging typically brings the correct peptide close to the top of the list of the high-scoring peptides (Table 4.4). This feature facilitates subsequent analysis of NRPs, for example, it allows one to correlate high-scoring reconstructions with NMR spectroscopy data. Moreover, the top-scoring peptide returned by NRP-Tagging typically have minor differences as compared to the correct peptide, for example, combining masses of adjacent amino acids or choosing a mass with known offset.

Lastly, we address a possible concern that NRP-Tagging may erroneously use a spectrum of a linear peptide to dereplicate a cyclic peptide. We show that if NRP-Tagging were to be run on a mass spectrum of a linear peptide, the resulting gapped peptides would have a much lower score than those of a spectrum of a cyclic peptide. Table 4.1 shows the top 5 tags for a spectrum of linear peptide Glu-1-Fibrinopeptide (glufib), a standard linear peptide used for instrument calibration, of sequence EGVNDNEEGFFSAR and the top 5 tags for tyrocidine C1. Both spectra were acquired using the same experimental settings. These results indicate

that NRP-Tagging may distinguish spectra from linear and cyclic peptides.

Table 4.1: NRP-Tagging results for linear peptide glufib and tyrocidine C1. % cuts is the percentage of observed fragment ions in the experimental spectrum out of all theoretical peaks in the cyclic peptide. % int is the percentage intensity explained by the annotated peaks in the spectrum (including possible neutral losses).

Glufib		
Sequence	**% Cuts**	**% Int**
65.95, 72.12, 646.54, 111.05, 342.09, 332.22	33	20
156.10, 115.00, 513.55, 542.20, 160.06, 83.05	30	19
64.94, 74.08, 643.78, 113.02, 341.13, 333.05	33	18
97.01, 132.05, 285.42, 114.04, 656.26, 285.18	37	16
57.02, 129.03, 757.67, 440.22, 114.98, 71.08	30	16
Tyrocidine C1		
Sequence	**% Cuts**	**% Int**
128.06, 146.04, 504.21, 283.12, 186.05, 114.03	50	58
128.06, 163.05, 487.20, 283.12, 186.05, 114.04	53	57
146.01, 99.07, 405.16, 283.13, 186.06, 242.12	53	54
128.06, 163.06, 99.02, 388.19, 283.12, 300.09	57	53
128.03, 163.06, 487.23, 300.10, 169.05, 114.05	50	52

4.3 Results

4.3.1 NRP-Dereplication

The results of the NRP-Dereplication algorithm are summarized in Table 4.2. The *Score* is defined as the product of the fraction of explained intensity and the fraction of explained fragment masses of a dereplicated peptide. Dereplicated matches have monomers (shown in red) where the candidate mutation is placed with the integer mass of the offset enclosed in square brackets in their monomer composition description. See Table A.1 for the complete list of monomers. Compounds that are in the database (tyrocidine A, B, C, H3526, microcystin LR and compound 879) or have a closely related compound (tyrocidines A1, B1, C1, cyanopeptide X, destruxin A) have higher scores than compounds that are not in the database and do not have closely related compounds (seglitide,

cyclomarin A, C and dehydrocyclomarin A, C). Dereplicated compounds have the mass difference of the experimental spectrum and the mass of the peptide enclosed in square brackets next to their name. The compounds are sorted by score in Table 4.2 and each dereplicated compound is in bold and separated by double horizontal lines. Compounds H8405 and BQ123 (representing the shortest peptides in the sample) returned incorrect matches (false positives). However, a close examination of the results revealed that these false positives are nevertheless correlated with the correct peptide sequences. For H8405, the correct sequences is [113, 71, 129, 186, 113], while the database match is [184, 186, 129, 113]. For BQ123, the correct masses are [113, 186, 115, 97, 99], while the database match is [71, 228, 71, 97, 143]. For seglitide and the family of cyclomarins, no high-scoring matches were returned by NRP-Dereplication because their sequences are not in NORINE yet. However, if we introduce any cyclomarin in NORINE, we readily dereplicate all other cyclomarins.

Table 4.2: NRP-Dereplication results. Each dereplicated compound is in bold followed by its top dereplicated matches with the name, score and dereplicated sequence. See the text for additional information.

Destruxin A	
Destruxin A[+14]	0.45
Pro, Ile, NMe-Val, NMe-Ala, bAla, C4:1(3)-OH(2)[+14]	
HydroxyDestruxin B[-18]	0.45
Pro, Ile, NMe-Val, NMe-Ala, bAla, iC5:0-OH(2.3)[-18]	
Destruxin D[-32]	0.45
Pro, Ile, NMe-Val, NMe-Ala, bAla, iC5:0-OH(2)-CA(4)[-32]	
Destruxin E diol[-20]	0.45
Pro, Ile, NMe-Val, NMe-Ala, bAla, C4:0-OH(2.3.4)[-20]	
Destruxin C[-18]	0.45
Pro, Ile, NMe-Val, NMe-Ala, bAla, iC5:0-OH(2.4)[-18]	
Destruxin F[-4]	0.45
Pro, Ile, NMe-Val, NMe-Ala, bAla, C4:0-OH(2.3)[-4]	
Destruxin B[-2]	0.45
Pro, Ile, NMe-Val, NMe-Ala, bAla, Hiv[-2]	
Destruxin E[-2]	0.45
Pro, Ile, NMe-Val, NMe-Ala, bAla, C4:0-OH(2)-Ep(3)[-2]	
Destruxin E chlorohydrin[-38]	0.45
Pro, Ile, NMe-Val, NMe-Ala, bAla, C4:0-OH(2.3)-Cl(4)[-38]	
Tyrocidine C	
Tyrocidine C	0.45
D-Phe, Pro, Trp, D-Trp, Asn, Gln, Tyr, Val, Orn, Leu	
Tyrocidine B[+39]	0.45
D-Phe, Pro, Trp, D-Phe[+39], Asn, Gln, Tyr, Val, Orn, Leu	
Tyrocidine D[-23]	0.45
D-Phe, Pro, Trp, D-Trp, Asn, Gln, Trp[-23], Val, Orn, Leu	

Table 4.2: NRP-Dereplication results. Continued from previous page

Tyrocidine B1	
Tyrocidine B[+14]	0.44
D-Phe, Pro, Trp, D-Phe, Asn, Gln, Tyr, Val, Orn[+14], Leu	
Tyrocidine C1	
Tyrocidine C[+14]	0.40
D-Phe, Pro, Trp, D-Trp, Asn, Gln, Tyr, Val, Orn[+14], Leu	
Tyrocidine A1	
Tyrocidine A[+14]	0.37
D-Phe, Pro, Phe, D-Phe, Asn, Gln, Tyr, Val, Orn[+14], Leu	
Tyrocidine B	
Tyrocidine B	0.37
D-Phe, Pro, Trp, D-Phe, Asn, Gln, Tyr, Val, Orn, Leu	
Tyrocidine A[+39]	0.37
D-Phe, Pro, Phe[+39], D-Phe, Asn, Gln, Tyr, Val, Orn, Leu	
Tyrocidine C[-39]	0.37
D-Phe, Pro, Trp, D-Trp[-39], Asn, Gln, Tyr, Val, Orn, Leu	
Tyrocidine A	
Tyrocidine A	0.33
D-Phe, Pro, Phe, D-Phe, Asn, Gln, Tyr, Val, Orn, Leu	
Tyrocidine B[-39]	0.33
D-Phe, Pro, Trp[-39], D-Phe, Asn, Gln, Tyr, Val, Orn, Leu	
Compound 879	
Neoviridogrisein	0.28
(Thr+Hpa), NMe-Ph-Gly, Ala, NMe-bMe-Leu, NMe-Gly, D-4OH-Pro, D-Leu	
H8405	
Beauverolide Ka[-18]	0.27
C10:0-Me(4)-OH(3), Trp, Phe[-18], D-aIle	
BQ123	
Halipeptin B[-20]	0.26

Table 4.2: NRP-Dereplication results. Continued from previous page

C10:0-Me(2.2.4)-OH(3.7), Ala, aMe-Cys[-20], NMe-OH-Ile, Ala	
H3526	
hymenistatin I	0.35
Pro, Tyr, Val, Pro, Leu, Ile, Ile, Pro	
hymenamide G	0.25
Pro, Tyr, Val, Pro, Leu, Ile, Leu, Pro	
Cyanopeptide X	
Majusculamide C[-30]	0.23
Map, Ala, Ibu, NMe-OMe-Tyr[-30], NMe-Val, Gly, NMe-Ile, Gly, Hmp	
Dolastatin 11[-30]	0.23
Gly, NMe-Val, NMe-OMe-Tyr[-30], Ibu, Ala, Map, Hmp, Gly, NMe-Leu	
Microcystin LR	
Microcystin LR	0.20
D-Ala, Leu, D-bMe-Asp, Arg, Adda, D-Glu, NMe-Dha	
[Dha7]microcystin-LR[+14]	0.20
D-Ala[+14], Leu, D-bMe-Asp, Arg, Adda, D-Glu, dh-Ala	
Microcystin LAib[+71]	0.19
D-Ala, Leu, D-bMe-Asp[+71], Aib, Adda, D-Glu, NMe-Dha	
Seglitide	
Microsclerodermin F[-3]	0.13
C12:3(7.9.11)-Me(6)-OH(2.4.5)-NH2(3)-Ph(12), Pyr[-3], NMe-Gly, D-Trp, Gly, OH-4Abu	
Cyclomarin C	
Aureobasin C[-60]	0.13
D-Hmp, NMe-Val, Phe, NMe-Phe, Pro, Val, NMe-Val, Leu, bOH-NMe-Val[-60]	
Cyclomarin A	
Aureobasidin F[-44]	0.12
D-Hmp, NMe-Val[-44], Phe, NMe-Phe, Pro, aIle, Val, Leu, bOH-NMe-Val	
Dehydrocyclomarin A	
Hymenamide J[-74]	0.12

Table 4.2: NRP-Dereplication results. Continued from previous page

Pro, Tyr, Asp, Phe, Trp[-74], Lys, Val, Tyr	
Dehydrocyclomarin C	
PF1022E[+44]	0.11
D-Lac, NMe-Leu, 4OH-D-Ph-Lac, NMe-Leu, D-Lac, NMe-Leu[+44], D-Ph-Lac, NMe-Leu	

4.3.2 NRP-Sequencing

The sequencing results of the NRP-Sequencing algorithm are summarized in Table 4.3. The reconstructed NRPs are represented as sequences of masses. For the sake of brevity, masses are rounded to integers. Composite masses (2 or more amino acids) are enclosed in square brackets. For example, [163+99] in tyrocidine A means that NRP-Sequencing returned 262 (composite mass of 163 and 99 (Tyr and Val)). Best reconstruction is the highest scoring completely correct (i. e. no incorrect *b*-ions) de novo sequence returned by NRP-Sequencing.

Table 4.3: NRP-Sequencing results. See text for discussion of the results.

Compound	Best reconstruction	Rank
Tyrocidine A	[163+99], 114, [113+147], [147+147], 147, [114+128]	1
Tyrocidine A1	[163+99], 128, [113+147], [147+147], 147, [114+128]	1
Tyrocidine B	[163+99], 114, [113+147], 97, [186+147], 114, 128	14
Tyrocidine B1	99, 128, [113+147], [97+186], 147, [114+128]	1
Tyrocidine C	113, 147, 97, 186, 186, 114, [128+163], [99+114]	125
Tyrocidine C1	[163+99], [128+113], 147, [97+186], 186, [114+128]	1
Seglitide	85, [163+186], 128, 99, 147	1
Cyanopeptide X	57, 113, 161, 141, 71, [113+114+57], 127	1
BQ123	113, 186, 115, [97+99]	1
H3526	97, [97+163], 99, [97+113], 113, 113	2
H8405	129, 71, 113, 113, 186	1

4.3.3 NRP-Tagging

The sequencing results of the NRP-Tagging algorithm are summarized in Table 4.4. The reconstructed NRPs are represented as sequences of masses. For the sake of brevity, masses are rounded to integers, e.g. NRP-Tagging reconstruction

for tyrocidine A is 99.06, 114.07, 113.07, 147.06, 97.05, 147.05, 147.05, 114.06, 128.03, 163.06, which is more accurate that the integer representation given in the first row of the Table. Composite masses (2 or more amino acids) are enclosed in square brackets. For example, [114+57] in cyanopeptide X means that NRP-Tagging returned 171 as the mass of an amino acid instead of the correct masses 114 and 57 (2-hydroxy-3-methyl-pentanoic acid and glycine). Incorrect masses are enclosed in curly brackets and expressed in terms of their offses from correct masses. For example, {97+1} in H3526 means that NRP-Tagging returned 98 while the correct mass is 97 (Pro). In this case the isotopic peak (rather than a *b*-ion) was chosen as the best spectral interpretation. Lastly, cases in which the algorithm splits a mass are enclosed in angle brackets with the correct mass followed by the masses returned by the algorithm. A single mass 286 in cyclomarin A is split as 129, 157. A single mass 222-18 (water loss) in compound 879 is split into 100 and 104. The reconstructions given in the table represent a complete reconstruction of the compound, or a reconstruction with composite masses and/or masses with a known offset. The "Best reconstruction" column presents the high-scoring peptide with a specified rank ("Rank column") that is selected from the list of all top-scoring peptides as the most similar to the correct peptide.

4.4 Discussion

Using mass spectrometry for NRP interpretation is a Catch-22 situation. On the one hand, there are no algorithms for interpretation of NRP spectra, thus providing little incentive for generating NRP spectra. On the other hand, shortage of NRP spectra slows down development of algorithms for NRP interpretation because spectral datasets are needed to develop such algorithms. Here we attempted to break this unfortunate cycle that will hopefully motivate the natural-product researchers to begin generating NRP spectra.

All software tools and spectral annotations described in the paper can be accessed at `http://bix.ucsd.edu/nrp/index.html`

Table 4.4: NRP-Tagging results. See text for nomenclature of the symbols used to annotate the sequences.

Compound	**Best reconstruction**	**Rank**
Tyrocidine A	99, 114, 113, 147, 97, 147, 147, 114, 128, 163	3
Tyrocidine A1	99, 128, 113, 147, 97, 147, 147, 114, 128, 163	16
Tyrocidine B	99, 114, 113, 147, 97, 186, 147, 114, 128, 163	4
Tyrocidine B1	99, 128, 113, 147, 97, 186, 147, 114, 128, 163	1
Tyrocidine C	99, 114, 113, 147, 97, 186, 186, 114, 128, 163	4
Tyrocidine C1	99, 128, 113, 147, 97, 186, 186, 114, 128, 163	1
Seglitide	85, 163, 186, 128, 99, 147	1
Cyanopeptide X	57, 113, 161, 141, 71, 113, [114+57], 127	1
BQ123	113, 186, 115, 97, 99	2
Destruxin A	113, 113, 85, 71, [98+97]	2
H3526	97, 97, 163, 99, {97+1}, 113, {113-1}, 113	10
H8405	129, 71, 113, 113, 186	2
Microcystin LR	{[83+71]+1}, {113-1}, {129-1}, {156+1}, 313, 129	27
Compound 879	113, 113, <222-18:100,104>, {147+18}, 71, 141, 71	7
Cyclomarin A	127, 139, <286:129,157>, 143, 71, [177+99]	10
Dehydrocyclomarin A	127, 139, 268, 143, 71, 177, 99	27
Cyclomarin C	127, 139, 270, {143+32}, {[71+177]-32}, 99	>40
Dehydrocyclomarin C	Not generated	-

4.5 Acknowledgements

This chapter, in full, was published as "Dereplication and de novo sequencing of nonribosomal peptides". J. Ng, N. Bandeira, W.-T. Liu, M. Ghassemian, T. L. Simmons, W. H. Gerwick, R. Linington, P. C. Dorrestein, and P. A. Pevzner. *Nature Methods*, vol. 6, pp. 596-599, 08 2009. Nuno Bandeira and the dissertation author were the primary authors of this paper.

Chapter 5

Dereplication of Noncyclic Peptides

Since the publication of the manuscript "Dereplication and de novo sequencing of nonribosomal peptides". J. Ng, N. Bandeira, W.-T. Liu, M. Ghassemian, T. L. Simmons, W. H. Gerwick, R. Linington, P. C. Dorrestein, and P. A. Pevzner. *Nature Methods*, vol. 6, pp. 596-599, 08 2009, additional developments have been implemented in the NRP-Dereplication algorithm described in Chapter 4.

5.1 Introduction

The results for cyclic peptide identification, especially those of database search (NRP-Dereplication), are very encouraging to extend the methods to noncyclic structures (Figure 5.1). In particular, partial cyclic structures represent the second most common class of nonribosomal peptides (NRPs) in Norine [66] (see Table 5.1). Being able to generalize the NRP-Dereplication algorithm will greatly help the natural product community to identify more new natural products (that are closely related to known ones).

Before generalizing NRP-Dereplication to noncyclic structures, it is worth describing existing methods of linear peptide identification and strategies to score a mass spectrum from a linear peptide. The essence of database search consists of finding the best match among all entries in the database. The database consists

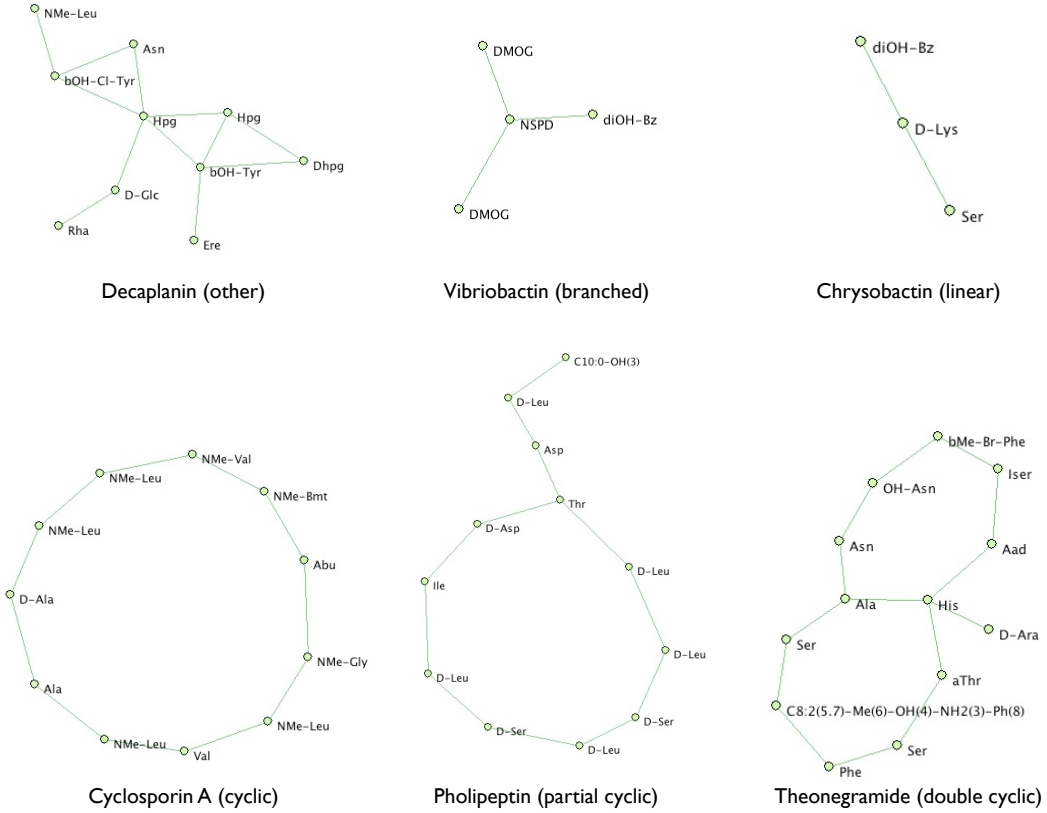


Figure 5.1: Examples of structures of NRPs in the Norine database.

Table 5.1: Distribution of NRPs in Norine according to their structure. The other class includes fairly complex compounds with many cycles and branches.

Structure	**Entries**
Cyclic	434
Partial Cyclic	281
Linear	254
Double cyclic	26
Other	71
Branched	5
All	1071

of sequence data, which does not have fragmentation information. Therefore, the strategy is to predict a theoretical spectrum from the sequence data, and compute a cross-correlation coefficient between the experimental and theoretical spectrum. With this approach, it is straightforward to find the best match, because one can create a theoretical spectrum for all peptide entries in the database and compute all the cross-correlation coefficients between the experimental and a theoretical spectrum. The critical step in database search is the ability to predict a theoretical spectrum that closely resembles a experimental spectrum, maximizing the cross-correlation coefficient (or score).

For linear peptides, the theoretical spectrum can be easily constructed by creating prefix and suffix fragments (b and y-ion for collision induced dissociation mass spectra). This approach is backed by the mobile proton fragmentation model [70, 71] and the great body of validated experimental data. The most naive approach for the creation of a theoretical experimental spectrum is simply creating a peak for each position corresponding a prefix and suffix ion. In here, the intensity of the peaks in the theoretical spectrum is ignored because all peaks have a uniform intensity. The most popular database search tool for mass spectra of linear peptides SEQUEST [1] uses this simple model with the addition of neutral loss peaks. This method works reasonably well in practice and is capable of identifying correctly spectra from linear peptides, but it is obvious that more reliable identifications can be obtained if the intensity information is used when creating the theoretical spectrum because not all fragments are equally likely to be created in an experimental spectrum. To this end, the main approaches for fragment intensity prediction rely on the fact that a correlation can be established between sequence data (amino acid composition, theoretical peak position, theoretical peak type, etc) and the intensity of the given fragment ion.

Zhang et al. [72] developed a kinetic model to generate theoretical spectra from sequence data. All fragmentation pathways can be modeled as a chemical reaction and a kinetic constant encapsulates the rate in which the reaction carries out. Under the kinetic model, all fragmentation pathways are in competition during the peptide fragmentation, and the intensity of a given peak is correlated

to the rate in which the pathway generating that peak is carried out. The set of fragmentation pathways can be compiled from the literature, and the kinetic constants can be derived from experimental data. Given a large corpus of validated data, this becomes an optimization/machine learning problem, in which the set of parameters (kinetic constants) need to be trained so that the maximum number of true positives are observed for the training set. For the intensity prediction problem specifically, the relative intensity of the peaks in the theoretical spectrum need to match those of the experimental spectrum.

The theoretical framework for peptide fragmentation is very complex, and not well understood. Although Zhang et al. [72] created a kinetic model by compiling known fragmentation pathways in the literature, the set of pathways considered is not complete or exhaustive. Therefore, other approaches do not try to explain all the mechanisms of fragmentation, but formulate the problem as a purely machine learning problem (optimize the parameters). The number of parameters required for a model to predict the peak intensities of a theoretical spectrum is typically quite large [73], indicating that the fragmentation process is very complex, but the advantage of formulating the problem in the framework of machine learning is that the mechanisms of fragmentation do not need to be understood.

5.2 Methods

The work in fragment prediction is directly applicable to the NRP-Dereplication algorithm. In fact, for linear NRPs existing database search tools can be readily applied with little or no modifications. However, for non-linear NRPs, the prefix-suffix fragment ion model is no longer applicable. To draw a parallel to database search of linear peptide mass spectra, the first step is to predict a theoretical spectrum that closely resembles the experimental spectrum of the peptide ignoring the peak intensities.

Theoretical Spectrum Prediction Problem. Given the primary sequence of a peptide $S = s_1, \ldots, s_n$ (series of monomers) and their connectivity $E = e_1, \ldots, e_m$ (e_i indicates a chemical bond between two monomers), return a list peak positions

$T = t_1, \ldots, t_l$ as a theoretical spectrum of S. Additionally, an input parameter b is needed to indicate the maximum number of breaks allowed in the original structure.

For linear peptides, the typical assumption is that $b = 1$. In fact, any prefix or suffix ion is the result of 1 break of the linear sequence. Cases of $b = 2$ are usually ignored because internal fragments account for a much smaller percentage of the total intensity of the spectrum. Any theoretical spectra created with $b > 2$ is no different than with $b = 2$ because no new peaks can be created if we allow more breaks on the structure.

For purely circular structures, the theoretical spectrum for $b = 1$ only has one peak with mass equal to the parent mass of the compound because for $b = 1$, the only event modeled is the linearization of the circular compound. For a mass spectrum to encompass sequence information, $b = 2$ needs to be used. Similar to theoretical spectra of linear peptides, for $b > 2$ does not add new peaks to the theoretical spectrum. Intuitively, experimental mass spectra from circular peptides require additional energy to break because we require to collect spectra with at least 2 breaks. Note that the algorithms described in Chapter 4 for the sequencing for circular NRPs make the assumption that there are two breaks in the spectrum.

To create theoretical spectra for peptides with more exotic structures (partially cyclic, branched and other) a combinatorial approach to generating candidate fragments needs implemented. For partially cyclic structures, it is reasonable to use $b = 2$ when generating the theoretical spectrum because this is the smallest b that encapsulates sequence information in the cycle part of the peptide. The creation of a theoretical spectrum with $b = 2$ is feasible because the maximum number of distinct fragments is strictly less than $(2 * \|E\| \ \textit{choose}\ 2) + 2 * \|E\| = \|E\|^2 + \|E\|$, where $\|E\|$ is the number of edges of the structure. For a NRP with 10 edges, the maximum number of theoretical fragments is less than 100. If neutral losses are considered, a multiplicative factor of the number of neutral losses needs to be factored into the calculations. $\|E\|$ *choose* 2 is the number of fragments that can be obtained with 2 breaks to the structure. The 2 coefficient is needed because for each configuration a complementary fragment is also generated with (potentially)

distinct mass. $\|E\|$ is simply the upper bound for the number of fragments that can be generated with a single break. For branched peptides, a theoretical spectrum constructed with $b = 1$ contains enough information to identify its peptide. Figure 5.2 shows fragment created from a partially cyclic and a branched NRP. Unlike linear and cyclic peptides, theoretical spectra of more complex structures add additional peaks for $b > 2$ adding complexity to the resulting theoretical spectrum.

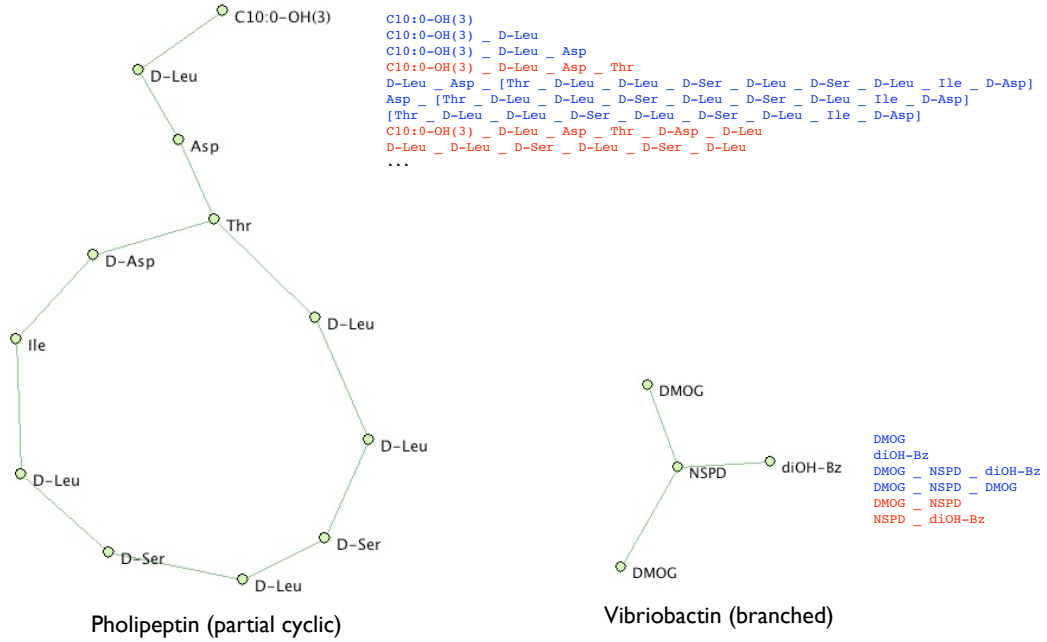


Figure 5.2: Theoretical fragment ions of the theoretical spectrum for 2 NRPs. The blue fragments are derived with one break while the red fragments are derived with two breaks. Not all theoretical peaks for pholpeptin are shown.

5.3 Results

The NRP-Dereplication for general structures was implemented and all peptides of the Norine database are considered in the search (as opposed to only cyclic structures in the previous implementation). In particular, viridogrisein (Figure 5.3) was initially included in the database by combining the monomers Thr and Hpa as a single unit, making the peptide completely cyclic. Compound 879 (Table 4.2) was dereplicated after adding viridogrisein to the database. However, with the

new NRP-Dereplication algorithm, this structure can be incorporated in its native form, and the additional fragments in the theoretical spectrum can be generated. However, no fragments missing the Hpa monomer were observed in the experimental spectrum (data not shown), meaning that no additional experimental fragments we explained with the new fragmentation model.

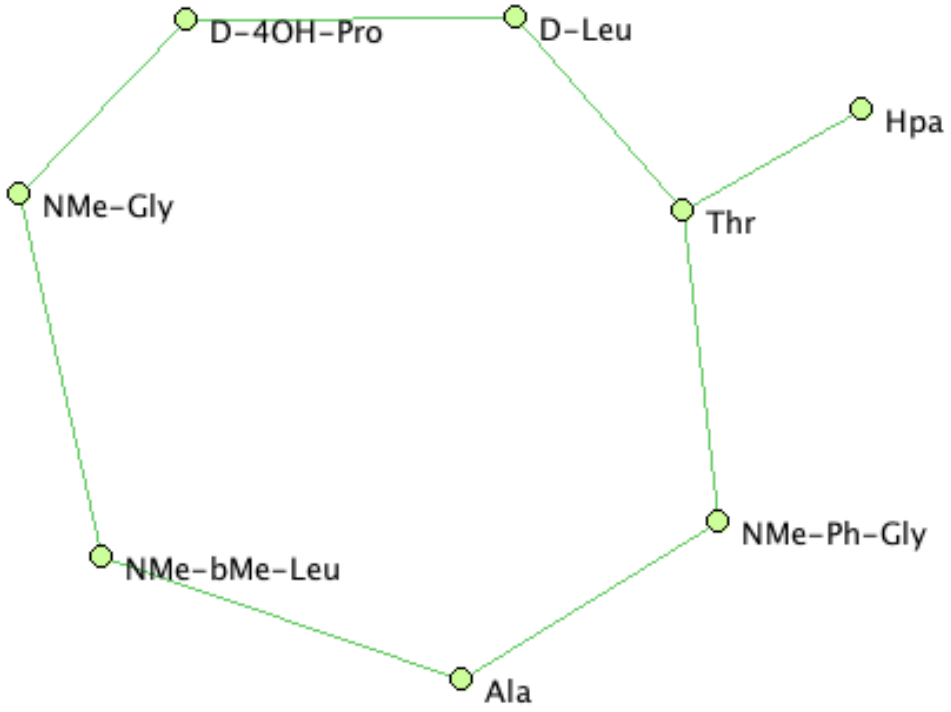


Figure 5.3: Structure of viridogrisein

5.4 Discussion

The proposed method for creation of theoretical spectra for general NRPs with different structures is not without shortcomings. Again, by drawing a parallel to database search of linear peptides, the proposed method uses a similar approach to SEQUEST. For instance, the peak intensity information is not used

when establishing the similarity between the experimental and theoretical spectrum. Two roadblocks still remain for the creation of more elaborated theoretical spectra. First, validated mass spectrometry data of NRPs is almost nonexistent in the field making the creation of statistical models impossible [31, 39, 74]. Second, the number of building blocks (monomers) is considerably higher for NRPs (vs 20 standard amino acid). This impedes using existing scoring models that predict peak intensity based on local amino acid composition of the fragment ion. However, a simple methodology (using uniform intensity) gave encouraging results for purely cyclic NRPs. Fortunately, the number of entries in Norine is quite small compared to protein databases used for linear mass spectrum searches. With accumulation of more mass spectrometry data from NRPs, we expect that more sophisticated models can be bootstrapped in the near future.

The generalized dereplication problem for arbitrary peptide structures can be directly extended to the framework previously described. In this case, we would exhaustively change the masses of the monomers in the peptide in the database and identify the modified version of the peptide that maximizes the similarity of the experimental and theoretical spectrum.

Finally, this approach for characterization of NRPs using mass spectrometry is not limited NRPs. Mass spectrometry data from polyketides, a group of secondary metabolites, could be analyzed given a database of polyketides.

Chapter 6

Interpretation of Tandem Mass Spectra Obtained from Cyclic Nonribosomal Peptides

Natural and non-natural cyclic peptides are a crucial component in drug discovery programs because of their considerable pharmaceutical properties. Cyclosporin, microcystins, and nodularins are all notable pharmacologically important cyclic peptides. Because these biologically active peptides are often biosynthesized nonribosomally, they often contain nonstandard amino acids, thus increasing the complexity of the resulting tandem mass spectrometry data. In addition, because of the cyclic nature, the fragmentation patterns of many of these peptides showed much higher complexity when compared to related counterparts. Therefore, at the present time it is still difficult to annotate cyclic peptides MS/MS spectra. In this current work, an annotation program was developed for the annotation and characterization of tandem mass spectra obtained from cyclic peptides. This program, which we call MS-CPA is available as a web tool (`http://lol.ucsd.edu/ms-cpa_v1/Input.py`). Using this program, we have successfully annotated the sequence of representative cyclic peptides, such as seglitide, tyrothricin, desmethoxymajusculamide C, dudawalamide A, and cyclomarins, in a rapid manner and also were able to provide the first-pass structure evidence of a newly discovered natural product based on predicted sequence. This compound

is not available in sufficient quantities for structural elucidation by other means such as NMR [75]. In addition to the development of this cyclic annotation program, it was observed that some cyclic peptides fragmented in unexpected ways resulting in the scrambling of sequences. In summary, MS-CPA not only provides a platform for rapid confirmation and annotation of tandem mass spectrometry data obtained with cyclic peptides but also enables quantitative analysis of the ion intensities. This program facilitates cyclic peptide analysis, sequencing, and also acts as a useful tool to investigate the uncommon fragmentation phenomena of cyclic peptides and aids the characterization of newly discovered cyclic peptides encountered in drug discovery programs.

6.1 Introduction

Ribosomally as well as nonribosomally derived cyclic peptides are an important group of compounds because of their wide range of biological, toxic, and pharmacological activities, and they often exhibit unique chemical structures [76, 77]. For example, the cyclic toxins microcystins and nodularins produced by cyanobacteria (blue-green algae) can wipe out entire fisheries and can cause death in humans [78, 79]. In addition, it is now becoming increasingly clear that these naturally occurring cyclic peptides have biological roles in quorem sensing [80, 81], gliding [82, 83], prevention of aerial growth [84], or cell adherence regulation [85] and that they can be used as a diagnostic markers for disease [86]. In addition, many cyclic peptides are used in the clinic. Well-known examples of cyclic natural products are cyclosporine, an immunosuppressant drug used to prevent organ rejection [87], seglitide, a potent growth factor release inhibitor [88], and ramoplanin, a novel antibiotic [89]. Because of the importance of their therapeutic applications, there is a continued development of strategies to generate cyclic libraries for drug screening programs [90, 91, 92, 93, 94]. In fact, many cyclic natural products with potent therapeutic properties are discovered every week [75, 95, 96, 97, 98]. Therefore it is important to continue developing methods not only for isolating or preparing such cyclic peptides but also to characterize such peptides.

Despite a lot of effort by mass spectrometrists [61, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110], we are still exploring the way cyclic peptides behave in a mass spectrometer, in particular during collision-induced dissociation (CID). Bioinformatics tools such as MASCOT, SEQUEST, and InsPecT are capable of robust interpretation of tandem MS spectra and also enable protein identification with the equipped database search engines [1, 2, 3]. However, few tools are designed for cyclic peptides with a user-friendly interface at a level accessible to non-mass spectrometrists. In addition most of the bioinformatics tools are based on somewhat refined fragmentation models, i.e., they may only annotate b and y ions. Both of these are the likely reasons why most scientists that isolate cyclic natural products and that develop cyclic peptide libraries for drug screening programs ignore all but only annotate a small amount of the ions that are typically observed from cyclic peptides in their structural elucidation efforts, leaving tens to hundreds of ions unaccounted for [103]. We became interested in this problem because when we attempted to annotate the tandem mass spectra of cyclic natural products isolated from marine organisms by manual means, we observed that a large proportion of the spectral intensity remained unaccounted for and that the annotation was very time-consuming. Although a program that predicts theoretical fragmentation patterns such as PFIA may assist in manual annotation of cyclic peptides by providing all possible b ions [111], MS-CPA is capable of direct annotation of the actual input cyclic peptide MS spectra and is also the first program that take into account the fragments that are a result of sequence-scrambling fragmentation pathways.

To improve our understanding of the fragmentation behavior of cyclic peptides we have developed a program that readily annotates a mass spectrum resulting from the collision-induced dissociation of cyclic peptides. In addition, we have created a user-friendly web interface so that other scientist that are noncomputer experts can easily use it to annotate their tandem mass spectra of cyclic peptides. Using this program, we observed that much of the spectral intensity of a MS^2 mass spectra of a cyclic peptide could not be explained. Upon further analysis, we realized that unanticipated fragmentation pathways were involved in cyclic peptides

when the standard fragmentation rules were applied. The data suggested those unanticipated fragments resulted in scrambling of the sequence. These unusual fragments were first described by Harrisons et al., as nondirect sequence (NDS) ions based on the scrambling of the original peptide sequence in contrast to the direct sequence (DS) ions derived from typical fragmentation pathways [103]. While initially surprising to the authors that NDS are observed, the mechanistic details toward the formation of NDS ions have recently been described in detail [107]. We have included NDS in our annotations. Therefore, our program, MS-CPA, not only provides evidence for the existence of these NDS ions but also enables quantitative analysis of the spectral abundance that match to DS and NDS ions.

In order to demonstrate the utility of this program, we have not only applied it to the representative testing peptides, seglitide and the tyrocidines, but also used it to confirm the sequence of two newly discovered natural products, desmethoxymajusculamide C (DMMC) and dudawalamide A, both isolated from marine cyanobacteria *Lyngbya majuscule* (Figure 6.1). In addition, the program was used to verify the structure of desprenylcyclomarin C, a natural product isolated from a prenyltransferase mutant of the marine bacteria *Salinispora arenicola CNS-205*. This marine natural product could not be isolated in sufficient quantities to confirm its structure by NMR; therefore, this program was critical in the confirmation of its structure. Finally, during these studies we discovered three additional dehydrated cyclomarin analogues and used our program to localize the site of dehydration.

6.2 Methods

6.2.1 Sample Preparation

Seglitide was purchased from Aldrich and was dissolved to a concentration of 20 μg/mL in 50:50 methanol (MeOH)/water with 1.0% acetic acid (AcOH). Dudawalamide A and DMMC were isolated from cyanobacteria and prepared in a solution of 50 μg/mL concentration in 50:50 MeOH/ water with 1.0% AcOH and was infused into the mass spectrometer. Cyclomarins were isolated from a marine

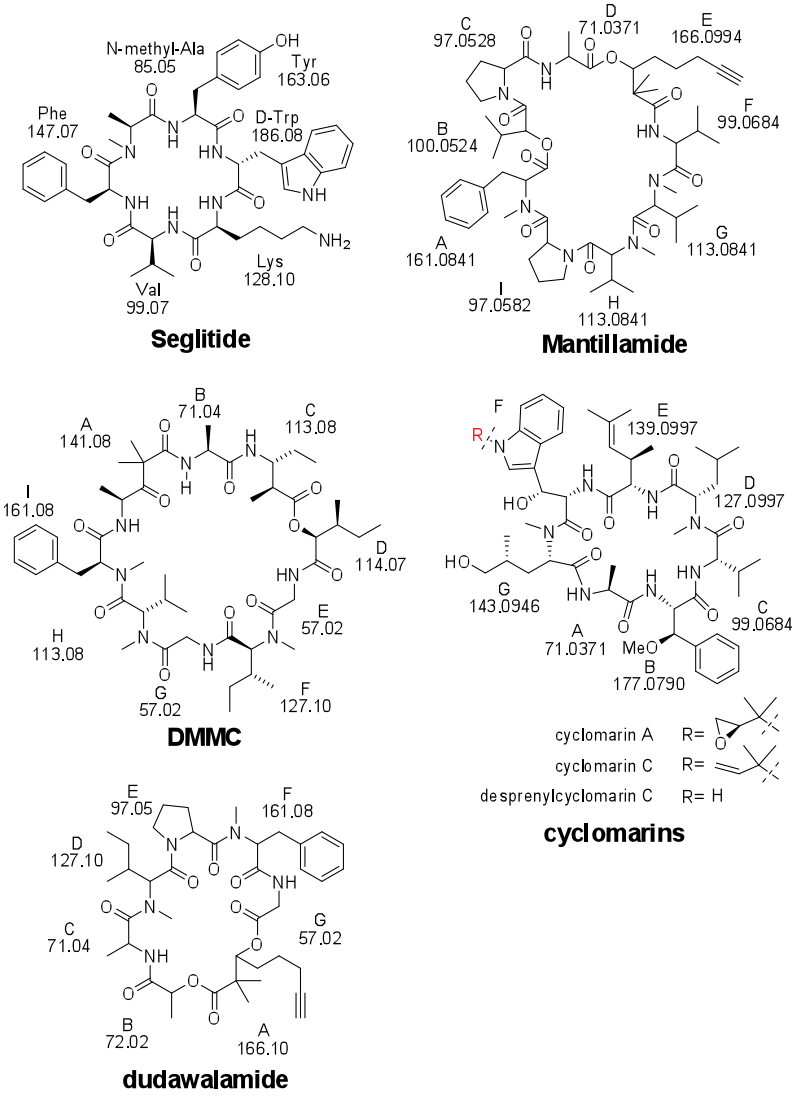


Figure 6.1: Structures of cyclic peptides discussed in this chapter.

actinomycete and desalted with C18 ZipTip pipet tips (Millipore) following the manufacturer's protocol to a final concentration of 50 μg/mL.

6.2.2 Mass Spectrometry

All samples were subjected to electrospray ionization on a Biversa Nanomate (Advion Biosystems, Ithaca, NY) nanospray source (pressure, 0.3 psi; spray voltage, 1.4-1.8 kV). Seglitide, tyrothricin, and DMMC were analyzed on a Finnigan LTQ-FTICR MS instrument (Thermo-Electron Corporation, San Jose, CA) running Tune Plus software version 1.0 and Xcalibur software version 1.4 SR1. Dudawalamide A was analyzed on a Thermo LTQ-Orbitrap-MS instrument (Thermo) running Tune Plus and Xcalibur software version 2.0. Activation time and q experiments, low-resolution spectra of seglitide, tyrothricin, and cyclomarins were acquired on a Finnigan LTQ-MS (Thermo- Electron Corporation, San Jose, CA) running Tune Plus software version 1.0. The final spectrum was obtained by averaging MS^2 scans with QualBrowser software version 1.4 SR1 (Thermo). Generally, the instrument was first autotuned on the m/z value of the ion to be fragmented. Then, the [M + H]+ ion of each compound was isolated in the linear ion trap and fragmented by collision induced dissociation (CID). Sets of consecutive, high-resolution, full MS/MS scans were acquired in centroid or profile mode and averaged using QualBrowser software (Thermo). The Thermo-Finnigan RAW files containing the average spectra were then converted to mzXML file format using the program ReAdW (tools.proteomecenter.org).

6.3 Results

6.3.1 Complexity of Cyclic Peptide Fragmentation

Because so many researchers work with cyclic peptides, the annotation of tandem mass spectra from cyclic peptides is important. The annotation, however, of tandem mass spectra of cyclic peptides is often difficult for mass spectrometrists and natural product scientists alike. The difficulty in the annotation of cyclic pep-

tides arises from the nature of cyclic peptides itself. A cyclic peptide with n amino acid residues, theoretically, will yield n series of b ions but not any y ions [105]. If there are other ions such as a ions, internal fragments, and small neutral losses such as H_2O and NH_3, this complexity increases significantly. Therefore, it is difficult to annotate each and every ion in the spectrum of cyclic peptides and thus becomes an informatics problem. To overcome some of the complexity in the annotation of these peptides, we have developed a program that assists in the annotation of tandem mass spectrometry data based on input amino acid values and an experimental tandem mass spectrometric data set in .dta and .mzXML formats.

While we have presented, at a conference, that de novo sequencing of these nonribosomal peptides can be accomplished with near perfect mass spectral data sets using spectral alignments and a combination of de novo and database searching algorithms [112], it quickly became clear that when we applied our first generation de novo sequencing algorithms to nonperfect mass spectrometry data sets typically encountered with more complex nonribosomally encoded peptides or symmetric cyclic peptides that these algorithms often identified a slightly different sequence. To improve the de novo sequencing algorithms that can be used to confirm the structures of isolated natural products, we need to improve our understanding of the resulting ions from a tandem mass spectrometry experiment. This is, in particular, important when it comes to complex cyclic peptides.

Cyclic Peptide Annotation Program. To aid in the sequencing as well as to improve our understanding of the fragmentation behavior of cyclic peptides of nonribosomal origin, we developed a program named the MS-Cyclic Peptide Annotation program (MS-CPA) that readily annotates a mass spectrum resulting from the CID of a cyclic peptide. In particular, this program annotates b ions, a ions (losses of CO), and b^0 ions (losses of H_2O). However, y ions are not included, because cyclic peptides do not yield such ions [105]. The annotation program started as a Python script to mark b, a, and b^0 ions given a mass spectrum. The current implementation is capable of handling .dta and .mzXML file formats as this data format is becoming the standard format for reporting or depositing

mass spectra and/or proteomic data sets [113, 114] as spectrum inputs. For the reason that many cyclic peptides contain unusual or modified amino acids, we leave the freedom for users to input the amino acid masses manually. There is no size limitation to the mass of the amino acid that can be manually imported. Additionally, default standard amino acids masses are provided. Finally, the amino acid sequence is specified by the user in the order that they are encountered in the peptide. For example, seglitide has a methylation on the nitrogen of alanine. This is a nonstandard amino acid; therefore, we can input 85.05280 for methyl-alanine rather than the alanine mass 71.03711. In addition, once it was recognized that even for mass spectrometrically well behaved peptides, a large proportion of the ion intensity remained unexplained, and the capabilities of this program was expanded to consider neutral amino acid losses from the b ion ladder as well as evaluation of possible rearrangements based on the series of masses initially given. The current program has thousands of lines of code to annotate a spectrum for the generation of a graphical and tabular output on a web server.

We have made the MS-CPA program publicly available as a web tool (`http://lol.ucsd.edu/ms-cpa_v1/Input.py`). In this work, we demonstrate the utility of MS-CPA for the characterization of the cyclic peptides shown in Figure 6.1. The cyclic peptides in Figure 6.1 are representative of the type of cyclic peptides encountered in drug screening programs.

6.3.2 Pre-analysis Data Processing of the Tandem Mass Spectrometry Input File

While the main code for this program is thousands of lines, the main challenge in the annotation process is actually the generation of a spectrum in which most peaks can be interpreted. Because of the great variance of experimental settings, instrumentation, and fragmentation properties of the compounds, preprocessing steps of the data that is required for each compound and experiment can vary a lot. To this end, we implemented a series of filters to enhance the signal-to-noise ratio of the experimental spectrum. Our current implementation regarding preprocessing includes centroid filtering, rank filtering, water filtering,

isotope filtering, peak tolerance, and symmetrization. Given that noise peaks are unavoidable in a real mass spectrometry experiment, the main goal of the filters is to eliminate ions that are likely noise or ions that are uninformative without losing the important data. In addition, this gives the users of this program the flexibility to annotate their spectra in a manner they prefer. For example, the user may only want to annotate the top 10 ions in the spectrum. This is possible with this interface. In addition it is possible to annotate unfiltered spectra but results in a much longer computational processing time. In many cases, in natural product research, the samples are available in limited quantities or the peptide does not fragment well and therefore it is not always possible to produce the best mass spectra. The filters will allow us to work with these spectra, instead of repeating the experiment, which might not be possible in real world drug discovery applications where there is often a limited supply.

6.3.3 Nomenclature of Ions

For discussion purposes of the results in this paper, we have adapted the nomenclature forwarded by Ngoka and Gross to describe the cyclic peptides in this paper [115]. The nomenclature developed by Ngoka and Gross describes the ions with a four-part descriptor with the general formula xnJZ, where x is the designation for the type of ion (b, a, etc.) and n is the number of amino acid residues that makes up the ion. J and Z are the one-letter codes for the two amino acid residues connecting the backbone amide bond, J-Z, which is broken to form the linear ion. J is the N-terminal amino acid residue and Z is the C-terminal amino acid residue. To illustrate the nomenclature, we use seglitide, a six-amino acid residue cyclic peptide illustrated in Figure 6.2 as an example. In seglitide and tyrocidines, the one letter amino acid abbreviation was used to represent each residue, while in other compounds we assigned letters in order of their sequence using the standard alphabet since they contained too many modified residues. For example, in this paper we describe DMMC for which 6 out of 9 are modified or nonstandard amino acids, while dudawalamide A has 4 out of 7 that are nonstandard, mantillamide has 5 out of 9, and cyclomarins have 5

out of 7 (Figure 6.1). Because the alanine in seglitide has methylation in the nitrogen position, we use A' to represent this methylated residue. Seglitide using this nomenclature would likely undergo random ring-openings following by the bn $\rightarrow$ bn-1 pathway45 resulting in the formation of 6 ($n = 6$) different series of b ions (Figure 6.2).

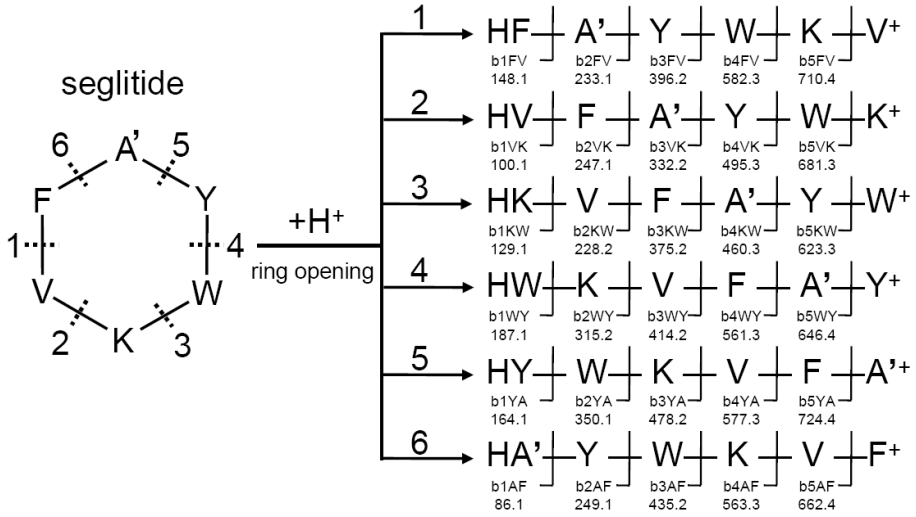


Figure 6.2: Schematic representation of ions in seglitide. According to the conventional pathway for fragmentation of cyclic peptides, seglitide first undergoes random ring-opening at each amide bond, yielding six different linear peptides. Sequential C-terminal amino acid cleavage results in six series of ions, for a total 30 b ions.

6.3.4 Cyclic Peptide Annotation Program on Seglitide

We first illustrate the application and utility of MS-CPA using a simple cyclic peptide, seglitide, a somatostatin receptor antagonist consisting of six amino acids, and described the results using the nomenclature defined above (Figure 6.1). Seglitide was analyzed by Fourier-transform ion cyclotron resonance mass spectrometry (FTICR MS). A singly protonated ion was observed at 808.4247 Da, which is within 3 ppm of the theoretical mass of seglitide (808.4272 Da). This ion was subjected to CID in a linear ion trap, and the product ions were again analyzed by FTICR MS (Figure 6.3). The resulting MS2 spectra were then ana-

lyzed by MS-CPA. The spectrum was subject to standard filtering procedures to increase the signal-to-noise ratio. First, because the raw spectrum was collected in profile mode, only the top peak was retained in a window of ±0.05 Da. Second, the top 200 most intense peaks were retained. Lastly, isotopic and water-loss peaks were filtered out, yielding 146 final peaks. As shown in Figure 6.4, the output of MS-CPA includes input residues and the parent mass that is obtained as user input or directly obtained from the input .dta or .mzXML file (A), summary of input filtering parameters and resulting ions counts (B), quantitative statistics of cleavage and total explainable ion intensity (C), a spectrum with color-coded matches (*b* ions are showed in red; water loss are green; *a* ions are cyan; NDS's are blue; unannotated ions are yellow) (D), a plot of mass errors of the annotated ions (E), and a list of matched fragment ions in tabular format (E).

For seglitide, the MS-CPA output indicates that 28 out of the 30 possible *b* ions were matched to observed masses. The explainable ion intensity of the *b* ions combined with possible a ions and loss-of-water ions was 71.5% of the total ion intensity. The absolute difference between the calculated and the experimental masses was less than 0.004 Da. Among the annotations, some of the ions with high intensity contained water loss even though there was no serine or threonine in the sequence. In addition, masses corresponding to addition of 28 Da (plus CO) were observed. These ions were not expected, thus we subjected these ions to additional rounds of tandem mass spectrometry (MS^3 and MS^4) to verify if the annotations were real or not. With these additional rounds of fragmentation, the authenticity of MS-CPA annotations was verified and these ions are indeed correctly annotated (MS^n spectra data not shown). Although the mechanisms behind the formation of these unusual fragments are still elusive, MS-CPA enabled us to discover the existence of these ions.

6.3.5 Observation of NonDirect Sequence Ions in Seglitide

Because more than 28% of the ion intensity remained unexplained, we explored the nature and significance of the remaining ion intensity. Because these data were acquired with high-resolution, the molecular mass of each ion could be

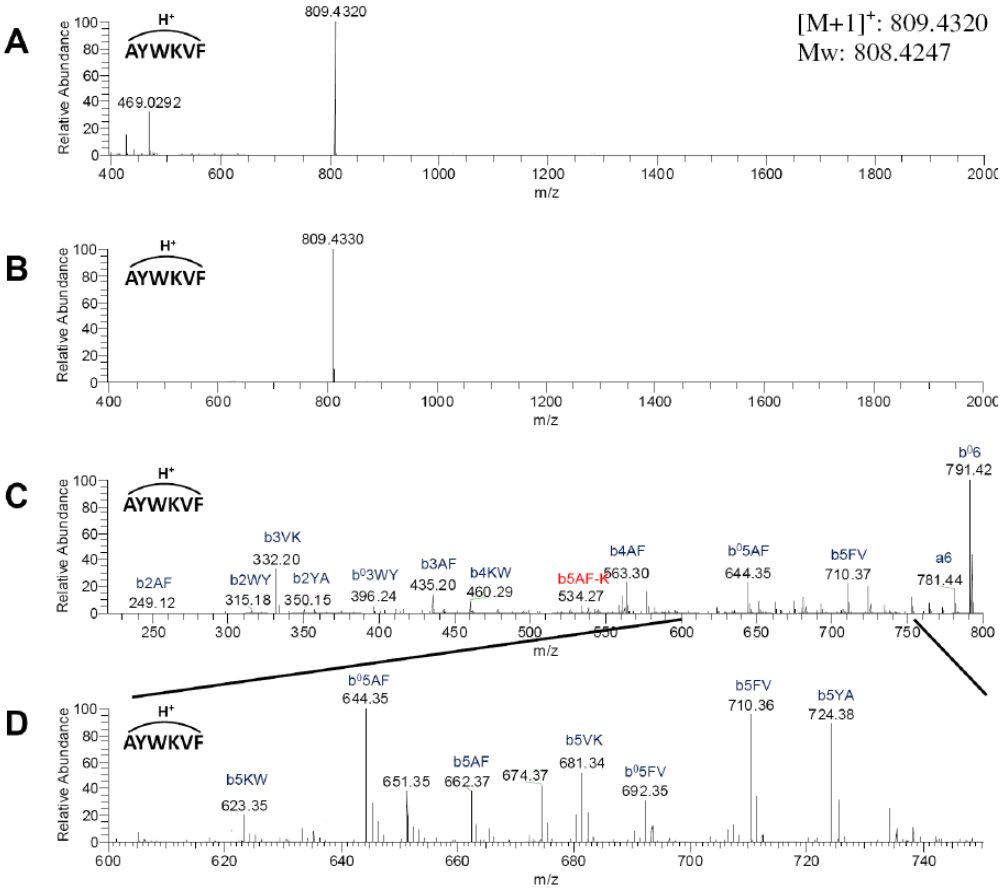


Figure 6.3: Seglitide MS and MS^2 spectrum. MS and MS^2 spectrum were collected by ESI-LTQ-FTICR MS. A) Broadband spectrum. B) Spectra obtained with an isolation window set for the seglitide parent ion (M + H)+. C) MS^2 spectrum of seglitide. D) Zoom in spectrum of the 600-750 m/z region.

A

Parameter	Value
Peak tolerance	0.006
Amino acids	A = 85.053 Y = 163.063 W = 186.079 K = 128.095 V = 99.068 F = 147.068
Modifications	-18.011 +18.011 -27.995 +27.995

B

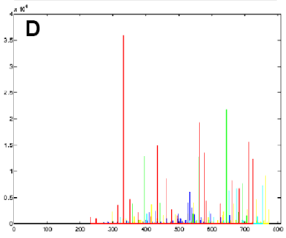
Filtering Method	Peak Count
Original	27912
centroid filtering width = 0.10 top = 1	547
Take top intensity peaks	200
Isotope filtering (optional)	146
Symmetrization	270

C

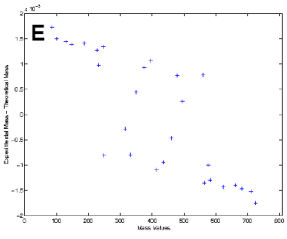
Description	Cuts	Fraction Int
A * Y * W * K * V * F	5/5	0.1418
F * A * Y * W * K * V	5/5	0.1452
V * F * A * Y * W * K	5/5	0.1322
K * V * F * A * Y * W	5/5	0.0852
W * K * V * F * A . Y	4/5	0.0557
Y . W * K * V * F * A	4/5	0.1545
Total explainable	28/30	0.7147
Total explainable by normal and GPR ions		0.8206

Experimental Parent Mass	808.4272
Theoretical Parent Mass	808.4272

D



E



F

top	mass	intensity	percent	annotation
1	332.197	35908.24	9.538	{332.197} VFA
2	644.354	21833.30	5.799	{644.356} AYWKV[-18.01]
3	563.297	19337.46	5.136	{563.298} AYWK
4	710.365	15516.38	4.121	{710.367} FAYWK
5	435.202	14833.12	3.940	{435.203} AYW
6	577.313	13561.04	3.602	{577.314} YWKV
7	396.239	12899.44	3.426	{396.240} WKV[-18.01]
8	559.302	12740.19	3.384	{559.303} YWKV[-18.01]
9	724.380	12244.54	3.252	{724.382} YWKVF
10	764.412	9253.73	2.458	{764.413} AYWKVF [-27.99]-[17.03]
11	460.291	8626.17	2.291	{460.292} KVFA
12	662.365	8156.27	2.166	{662.367} AYWKV
13	692.354	7673.47	2.038	{692.356} FAYWK[-18.01]
14	752.375	7271.79	1.932	{752.377} YWKVF[+27.99]
15	681.339	6730.78	1.788	{681.340} VFAYW
16	674.365	6699.22	1.779	{674.367} WKVFA[+27.99]
17	651.349	6295.16	1.672	{651.351} KVFAY[+27.99]
18	534.270	5998.01	1.593	{534.272} AYWV
19	734.365	4694.99	1.247	
20	350.150	4640.29	1.233	{350.150} YW
21	582.270	4379.05	1.163	{582.272} FAYW
22	544.291	3995.58	1.061	
23	527.297	3928.30	1.043	{527.298} AWKV[+27.99]
24	442.281	3905.33	1.037	{442.282} KVFA[-18.01]
25	623.354	3787.63	1.006	{623.356} KVFAY
26	357.228	3757.14	0.998	{357.229} KVF[-18.01]
27	665.344	3747.00	0.995	
28	416.229	3646.87	0.969	
29	315.181	3521.48	0.935	{315.182} WK
30	538.302	3083.50	0.819	{538.303} YKVF

Figure 6.4: MS-CPA output from analysis of seglitide MS^2 data. A) and B) MS-CPA input parameters summary, number of cleavages, and explainable intensity. C) Annotation; the * indicates an ion that cleaves here was annotated in the spectrum. D) Annotated spectrum. E) Accuracy analysis. F) Annotated ions list (unsymmetrized) (to save space, only the top 30 intensity ions were displayed). In the output spectrum and annotation list, the *b* ions are showed in red; H_2O loss is green; *a* ions are cyan; NDS's are blue; and unannotated ions are yellow in the spectrum and unlisted in the table. Symmetric ions are not shown in parts D or F.

determined. First, we analyzed these for alternate combinations of amino acids that would result from peptide residues rearrangements. We found 58 such ions comprising roughly 10% of the total ion intensity. Each of these scrambled sequence ions had mass errors within 0.004 Da, in agreement with all of the other masses we had annotated. The fact that so many of the ions could be explained by a rearrangement of the amino acid sequence is unlikely be coincidental or due to noise. In fact, some of these scrambled ions are of relatively high abundance. In seglitide, the most abundant NDS ion was up to 16% of the normalized ion intensity when the most intense ion was set to 100%. These kinds of scrambled sequence ions have previously been observed in peptides and described as nondirect sequence ions [101, 103, 104, 105, 107, 108, 109, 110]. Because of their relatively high abundance, they are included into our annotation program MS-CPA. By their inclusion, the accountable signal intensity increases from 71.5% to 82.1%. Notably, some ions still remain unannotated, these ions are likely a result of side chain fragmentations, unknown fragmentations, or noise inherently present in the mass spectrometry data set.

To confirm the presence of NDS ions from seglitide, the two most intense of these ions, AYWV and YKVF ($b_{5\mathrm{AF}}$-K, $b_{5\mathrm{YA}}$-W), and each $b5$ ion (i.e., the parent ion minus one amino acid) were isolated and subjected to an additional round of CID. The $b5$ ions were chosen for comparison and were anticipated to be linear by conventional fragmentation pathways $b_{\mathrm{x}} \rightarrow b_{\mathrm{x\text{-}1}}$ [116]. Surprisingly, the MS^3 spectra indicated that none of these selected ions simply followed the conventional rules for fragmentation which state that cyclic peptides sequentially lose amino acid residues from the C-terminus after the initial ring-opening event (Figure 6.5) [99]. Instead, we observed a mixed series of b ions (Figure 6.2) which suggest that the precursors for the MS^3 experiment are still cyclic. For example, if the $b5$ ion FAYWK was of linear structure, only the $b_{n\mathrm{FK}}$ ion series should be present in the associated MS^3 spectrum (Figure 6.5A); however, we observed the relatively intense $b_{n\mathrm{YA}}$ and $b_{2\mathrm{WY}}$ ions. These additional ion fragments most likely originate from cyclic peptide precursors.

To explore these NDS ions behavior, we first compared the total ion inten-

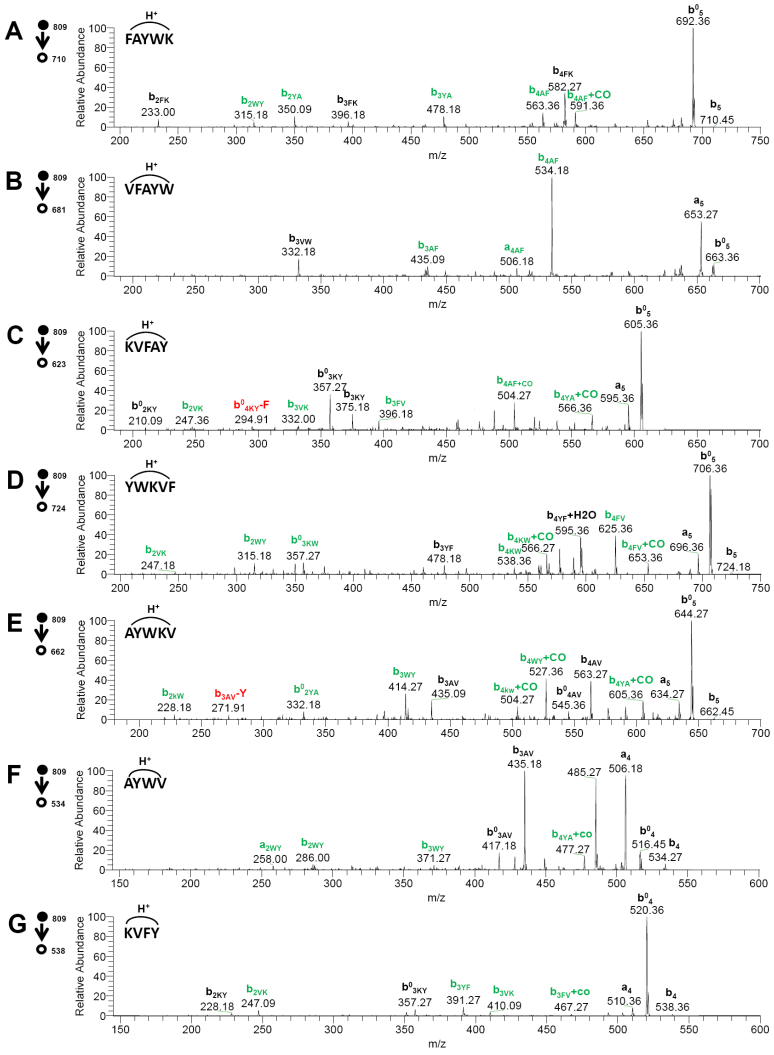


Figure 6.5: MS^3 spectra of representative seglitide sequence ions. The presence of daughter *b* ions from different linearized parent ions suggests that the parent ion is cyclic (as opposed to linear, as initially assumed). MS^3 spectra were collected by ESI-LTQ MS. A-E) *b*5 ions. F) and G) top two NDS ions observed. Expected sequence ions of a linear peptide are shown in black. The expected ions for the cyclic peptide are showed in green combined with the black ones. The red represents NDS ions.

sities explained by assuming a linear precursor with those explained by assuming a circular precursor. For example, CID on the ion $b_{5\mathrm{KW}}$ (KVFAY) would yield K, KV, KVF, KVFA, Y, AY, FAY, and VFAY fragments if the $b_{5\mathrm{KW}}$ ion was linear. However, if this ion was circular, we would observe 20 possible fragments. In the case of $b_{5\mathrm{KW}}$ (Figure 6.5C), the ions annotated as AYKV, FAYK, and YKVF show high intensity and are easily explained if the precursor ion is considered to be circular. In fact, 88% of total ion intensity can be explained by assuming a circular precursor, while only 42% can be explained by assuming a linear precursor. Table 6.1 summarizes the analysis of the seven MS^2 ions that were subjected to additional CID and annotated as either linear or circular. Among these seven MS^2 ions, the only one that gave poor fragmentation is $b_{5\mathrm{VK}}$ (VFAYW), with 12 cleavages out of 20. However, this ion produced a very intense peak ($b_{4\mathrm{AF}}$) that corresponds to loss of phenylalanine. This peak would not have been the most intense ion in the MS^3 spectrum if the initial cyclic peptide had first undergone linearization and then eliminated the C-terminal residue (i.e., tryptophan) as predicted by conventional fragmentation rules [99, 115]. While all of the foregoing results strongly support the cyclic nature of the MS^2 ions resulting from CID of seglitide, it is likely that a mixture of cyclic and linear forms ultimately contribute to the MS^3 spectrum.

Table 6.1: MS-CPA analysis of the two most intense NDS ions and $b5$ ions of seglitide. Results were analyzed by isotope removal, water removal, NH_3 removal, and window filtering with width 10 Daltons and top 10 peaks. The fragments column represents the fragments that cover the linear breakpoint.

Ion	Linear		Circular		Fragments
	cuts	intensity	cuts	intensity	
b5FV	6/8	64.75%	13/20	89.24%	3
b5VK	5/8	20.46%	12/20	88.82%	4
b5KW	4/8	42.39%	11/20	88.36%	4
b5YA	6/8	46.15%	14/20	92.54%	6
b5AF	5/8	55.19%	8/20	82.01%	1
b5AF-K	4/6	73.16%	7/12	85.38%	2
b5YA-W	4/6	50.47%	8/12	76.81%	3

Although the formation of these NDS ions have been recognized since 2003 [103], the actual mechanisms behind them are still a hot research topic.

Several groups have argued the importance of understanding this phenomenon in the development of de novo sequence programs. Therefore, a few mechanisms have been proposed to account for NDS ions [101, 103, 104, 105, 107]. The general consensus involves a cyclic intermediate occurring by recyclization. The presence of recyclized intermediates have been verified by Riba-Garcia and co-workers using ion-mobility MS [109, 110]. The tendency of generating NDS ions was also studied under N-acetylation modification or various activation energies [108]. Recently, just after this current manuscript was submitted, a more thorough mechanism and pathway was published by Bleiholder et al., in which a sequence-scrambling fragmentation pathway was proposed describing the mechanism of NDS ions based on experimental and energetic calculations in agreement with the cyclic NDS ions we observed [107]. Therefore, our program, MS-CPA, provides solid evidence showing the existence and abundance of these NDS ions with nonribosomally derived cyclic peptides.

6.3.6 Capability of MS-CPA in Analyzing an Antibiotic Mixture

In addition to seglitide, we investigated the antibiotic mixture tyrothricin, which contains more than 28 different compounds and is readily available commercially due to its clinical utility as a typical antibiotic. Some of these compounds, individually called tyrocidines, are known to be cyclic peptides [117]. We used MS-CPA to analyze several ions from this mixture (Figure 6.6). In the case of tyrocidine A, the program successfully annotated 74 b ions out of 90 possible. In contrast, only 17 b ions were identified through manual annotation of tandem mass spectra from tyrocidine A, despite this being one of the most thorough studies of cyclic peptides available to date in the literature demonstrating a significant advantage of spectra using our approach [118].

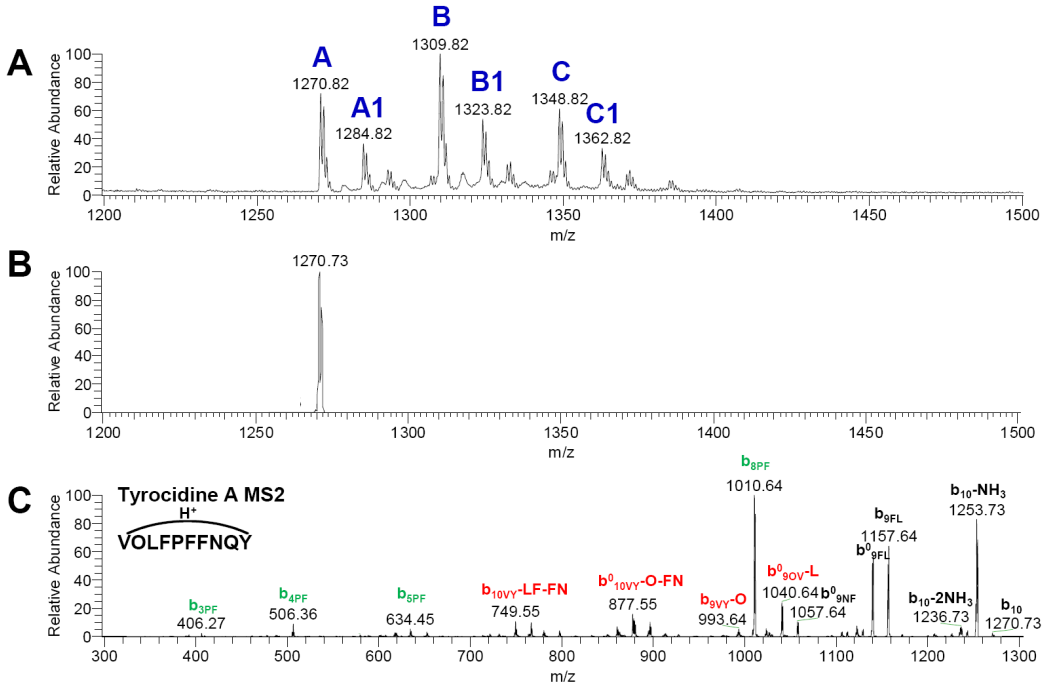


Figure 6.6: Tyrocidines MS and MS^2 spectra. MS and MS^2 spectra were collected by ESI-LTQ MS. A) Broadband spectrum showing different species of tyrocidines in tyrothricin antibiotic mixture. B) Isolation of tyrocidine A (protonated form). C) MS^2 spectrum of tyrocidine A.

6.3.7 Using MS-CPA to Annotate Cyclic Peptides Containing Nonstandard Subunits

Seglitide and the tyrocidines have a uniform peptidic backbone with standard amino acids. However, many nonribosomal cyclic peptides are cyclized via lactone formation and include nonstandard amino acids [119]. Theoretical calculations suggested that cyclic peptides favor a lactone bond as the initial ring-opening site, and also the fragmentation pathway of cyclic peptides differs when lactone bond(s) were involved [102]. It is therefore important to establish how these other structural features impact the fragmentation data and the results analyzed by the MS-CPA program. Thus, we analyzed several nonribosomal cyclic peptide natural products containing lactone linkages and nonstandard amino acids by tandem MS followed by MS-CPA (Table 6.2). These included three marine cyanobacterial depsipeptides: desmethoxymajusculamide C (DMMC), mantillamide, and dudawalamide A, all three of which were isolated because of their biological activity to cancer cells or malaria parasites (Figure 6.1). Analysis of DMMC by MS-CPA uncovered 36 of the 72 *b* ions expected from standard fragmentation. Including NDS ions, the proportion of explained total ion intensity increased from 71.1% to 78.3%. Similar results were obtained for mantillamide. These data indicate that nonstandard residues and ester linkages do not diminish the program's ability to insightfully annotate a tandem mass spectrum.

Dudawalamide A was isolated from the marine cyanobacterium *Lyngbya majuscula*, and its structure was determined by NMR methods. A high-resolution MS^2 spectrum of this compound was submitted to MS-CPA for annotation. The program was also provided with the masses of the dudawalamide subunits determined by NMR. The fragmentation behavior of dudawalamide, also a lactone, was found to be very different from the fragmentation behavior of mantillamide and DMMC. Although 96.0% of the total ion intensity was explained by *b* ions with absolute mass errors smaller than 0.008 Da, only 18 of the predicted 42 *b* ions were identified by the program. Thus, a high proportion of total ion intensity was accounted for by a small fraction of the expected *b* ions. This phenomenon can be explained by the presence of labile connections between residues within du-

dawalamide. Such weak connections are represented in normal peptides by amides N-terminal to prolines, amides C-terminal to Asp and Glu, or amides involving tertiary amines [101]. Three such linkages are present in dudawalamide: one at the N-terminus of proline and the other two at the N-termini of the N-methylated phenylalanine and the N-methylated isoleucine. Because of these three labile connections, the fragmentation of dudawalamide produced only a few ions, which were consistent with the known structure of dudawalamide but provided little sequence coverage.

Lastly, we used MS-CPA to investigate the structures of cyclomarin A, cyclomarin C, and desprenylcyclomarin C. The natural products cyclomarin A and C were originally isolated, based on their strong anti-inflammatory activity, from the marine bacterium *Streptomyces sp. CNB-982* [120]. Subsequently, desprenylcyclomarin C was isolated from a prenyltransferase mutant of *Salinispora arenicola CNS-205*, but could not be produced in amounts sufficient to enable structural characterization by NMR [75]. We therefore subjected all three cyclomarins to mass spectrometry and acquired MS^2 spectra of each analogue. The broadband mass spectra of each of these cyclomarins showed a protonated ion species and a even much more stronger species corresponding to dehydrated forms (data not shown), providing evidence that these natural products are prone to water loss. The MS^2 spectra of both the protonated and dehydrated forms of each cyclomarin analogue were collected and subjected to MS-CPA.

Analysis by MS-CPA consistently revealed the presence of strong b_{5GF} and b_{4AG} ions in the MS^2 spectra of all of these cyclomarin species (Table 6.2), thus confirming that desprenylcyclomarin C is structurally related to cyclomarin A and C. Overall, these analyses of cyclomarins identified from 8 to 16 b ions out of 34 possible b ions. The fraction of explained total ion intensity ranged from 37.0 to 50.3% when NDS ions were excluded and from 47.2 to 72.5% when NDS ions were included. On the other hand, this fraction was much higher for the dehydrated forms of cyclomarins, ranging from 72.2 to 79.1% without NDS ions and from 76.4 to 91.8% with NDS ions. In addition, we have successfully localized the dehydration site to the tryptophan-derived residue. Because cyclomarins are

so prone to dehydration, it is possible that this is the form that provides its anti-inflammatory activity. The most likely path leading to dehydration is the formation of an imine on the tryptophan residue, yielding a conjugated system upon loss of water. These examples highlight the usefulness of MS-CPA to assist in the structural characterization of cyclic nonribosomally encoded natural products even when limited quantities are available.

6.4 Discussion

Because cyclic peptides are an important class of therapeutics and toxins, we have developed a program, MS-CPA, to facilitate the structural characterization of these types of natural products. Users can easily access the program on the World Wide Web in order to annotate their tandem mass spectra of cyclic peptides. Using this program, we solidified the amino acid sequence of several recently discovered bioactive natural products, such as dimethoxymajuscalide (DMMC), mantillamide, dudawalamide A, and verified the structure of desprenycyclomarin C as well as dehydro-desprenylcyclomarin C that were isolated from a des-prenyltransferase knockout *S. arenicola CNS-205* strain. This analysis demonstrates the strength of this program when combined with tandem mass spectrometry, as well as a candidate structure enables the structural characterization of cyclic peptides produced in such low quantities that normally prohibit the use of other structural methods such as NMR.

Using our annotation program, we observed that cyclic nonribosomal peptides fragment in unusual ways. This kind of sequence-scrambling fragmentations results in a spontaneous recyclization event. The observation of NDS ions makes the problem of de novo sequencing of cyclic peptides even more challenging than was previously anticipated. Therefore, the annotation and understanding of the fragmentation patterns will, undoubtedly, facilitate and improve de novo sequencing algorithm development.

In summary, our current developed program provides a rapid annotation platform for tandem MS spectra of cyclic peptides. Also, although not designed

for this, it can likely also be used to analyze the cyclization phenomenon of linear peptides. We are currently using this program to annotate peptides that have been isolated from marine organisms that have potent cancer, malarial, and antibiotic resistant bacterial inhibitory activities. The approach described in this paper should be useful to the studies of cyclic peptide virulence factors, the chemical ecology of cyclic peptides, as well as cyclic peptides in drug screening programs [75, 121, 122, 123, 124].

6.5 Acknowledgements

This chapter, in full, was published as "Interpretation of tandem mass spectra obtained from cyclic nonribosomal peptides". W.-T. Liu, J. Ng, D. Meluzzi, N. Bandeira, M. Gutierrez, T. L. Simmons, A. W. Schultz, R. G. Linington, B. S. Moore, W. H. Gerwick, P. A. Pevzner, and P. C. Dorrestein, *Analytical Chemistry*, vol. 81, no. 11, pp. 4200-4209, 2009. Wei-Ting Liu and the dissertation author were the primary authors of this paper.

Table 6.2: Summary of MS-CPA analysis of cyclic peptide natural products. "des" represents Desprenyl. "dehy" represents dehydrated. For the cyclomarins, manual annotations for ions reflecting loss of methanol were included in the calculations of explainable ion intensity.

Name	Cuts	Explained intensity	
		Without NDS ions	With NDS ions
DMMC	36/72	71.10%	78.30%
Dudawalamide A	18/42	96.00%	97.30%
Mantillamide	34/72	65.21%	81.99%
Cyclomarin A	12/42	50.34%	72.48%
Cyclomarin C	16/42	36.98%	47.20%
Descyclomarin C	8/42	46.74%	55.48%
Dehycyclomarin A	16/42	73.79%	87.88%
Dehycyclomarin C	12/42	76.35%	91.83%
Dehydescyclomarin C	12/42	72.16%	79.09%

Chapter 7

Web Interface for Annotation and Interpretation of Cyclic Peptides

7.1 Introduction

The tools described for annotation and analysis of cyclic peptides can be accessed and run through our webserver (accessible via `http://proteomics.ucsd.edu/`). The main advantage of providing a webserver to run the tools is that, as a developer, one is freed from deploying the software on multiple platforms. However, significant efforts were invested into creating a web interface for a friendly and intuitive user-experience. A snapshot of the current version of the webserver is illustrated in Figure 7.1.

The work of developing a webserver was inspired by the philosophy that the user should be able to do all the analysis with a web browser. The model of delivering an executable to the end user (as opposed to making the software runnable through a web interface) has prevented most biologists from installing the software, let alone running the software. Hosting the software in a server, also has the advantage of having the most current version available to the user. Bugs fixes and updates can also be issued immediately to the version of the software being hosted in the webserver.

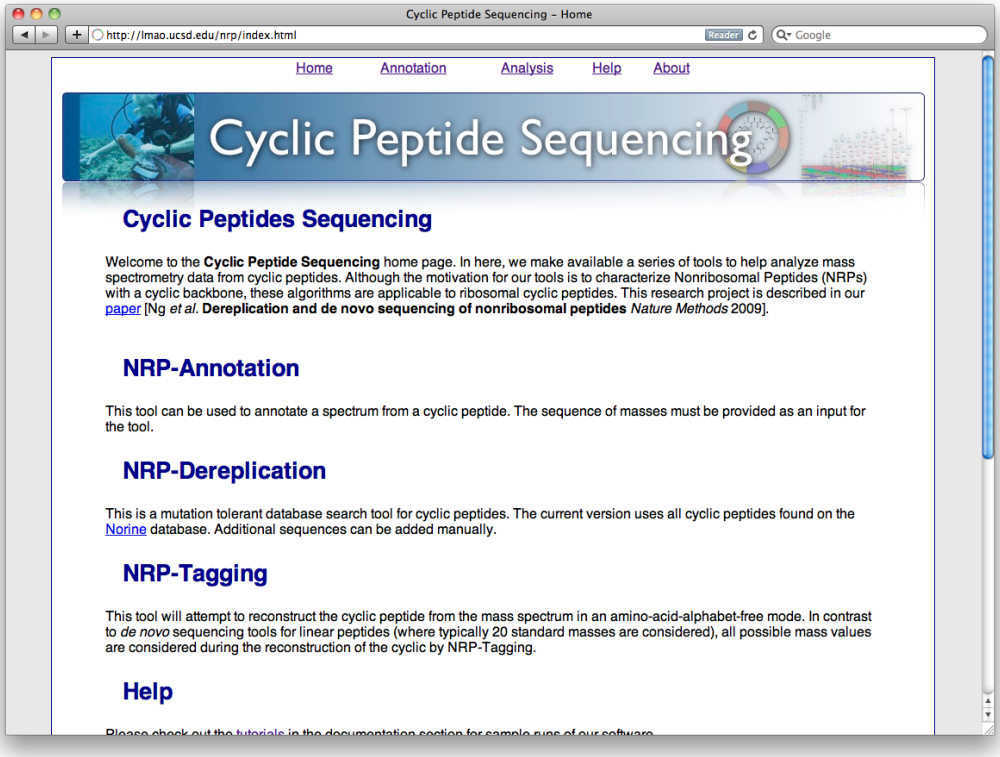


Figure 7.1: Snapshot of the homepage of the cyclic peptide software tools.

7.2 Methods

The webserver is implemented using the Common Gateway Interface (CGI) script module of the Python programming language. The webserver has two main workflows. The first workflow called NRP-Annotation, annotates a spectrum in a circular fashion [125] given a sequence of amino acids. The second workflow is a multistep called NRP-Analysis, which analyses the spectrum using the NRP-Dereplication and NRP-Tagging algorithms [125].

7.2.1 NRP-Annotation

The main page for the NRP-Annotation algorithm is presented in Figure 7.2. The user is asked to input the name of the run, the tolerance used for the annotation, an email for job completion notification, the file in mzXML or dta format and the peptide sequence. Once the file is uploaded, the user has the option to correct the parent mass manually by filling the form once the spectrum file is uploaded. If the uploaded file is in mzXML format, the user can choose the spectrum to be annotated in the XML tree (see Figure 7.3). Once the job is submitted the user is redirected to an auto-refreshing transition page that will direct the user to the result page. The result page (not shown) contians a series of tables that contain information about the number of breaks and percentage of annotated intensity for the spectrum. A sortable peak list is also shown with candidate annotations given the input amino acid masses and tolerance. A custom JavaScript (`http://bix.ucsd.edu/projects/files/sorttable.js`) was written to manipulate and sort the elements of the table.

In terms of user-friendly visualizations, a spectrum image is generated illustrating the circular annotations and relative intensity of the annotation (see Figure 7.4). The image is generated with the help of the Python graphics package (PyX). The resulting image is a high quality vector graphics image that is publication ready. All the drawing software was written from ground up to provide maximum flexibility in the placement of the annotated elements. The code for the drawing portion of this project contains code from rendering the spectrum to

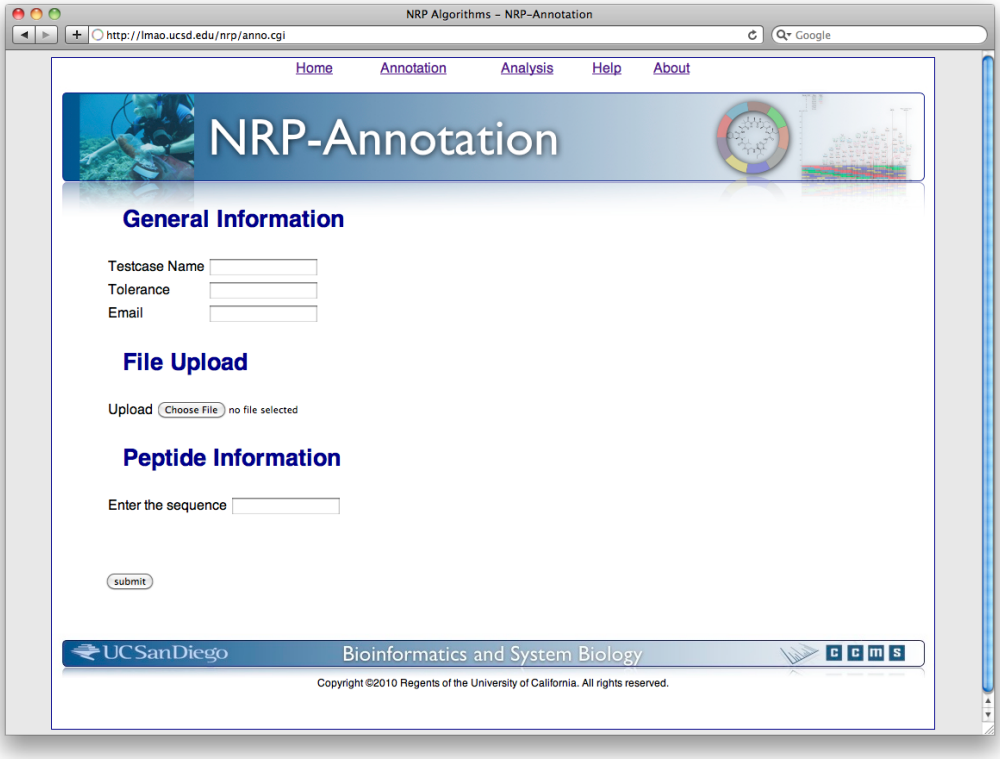


Figure 7.2: Snapshot of the main page of the NRP-Annotation.

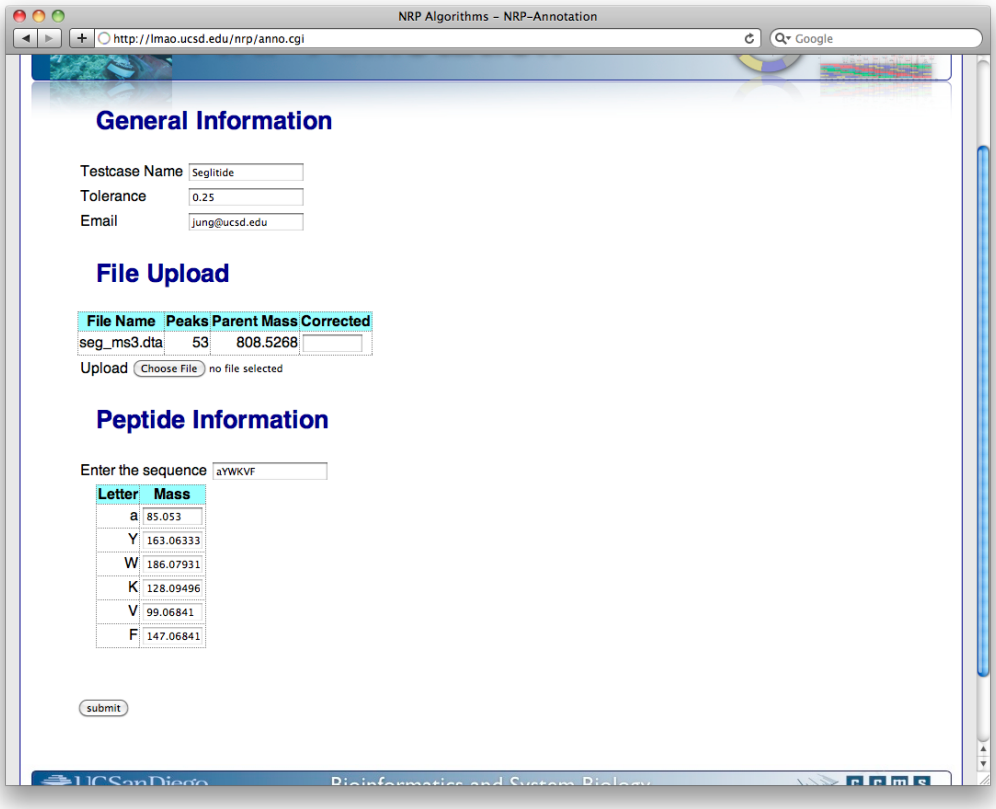


Figure 7.3: NRP-Annotation input parameters. The sample input parameters are shown for seglitide. Standard amino acid masses are automatically pre-filled according to their single uppercase letter representation. Nonstandard amino acids (lower case letters) need to be inputted manually. The user can force the program to use a specific parent mass for the spectrum by filling in the input form.

placement of the annotation labels.

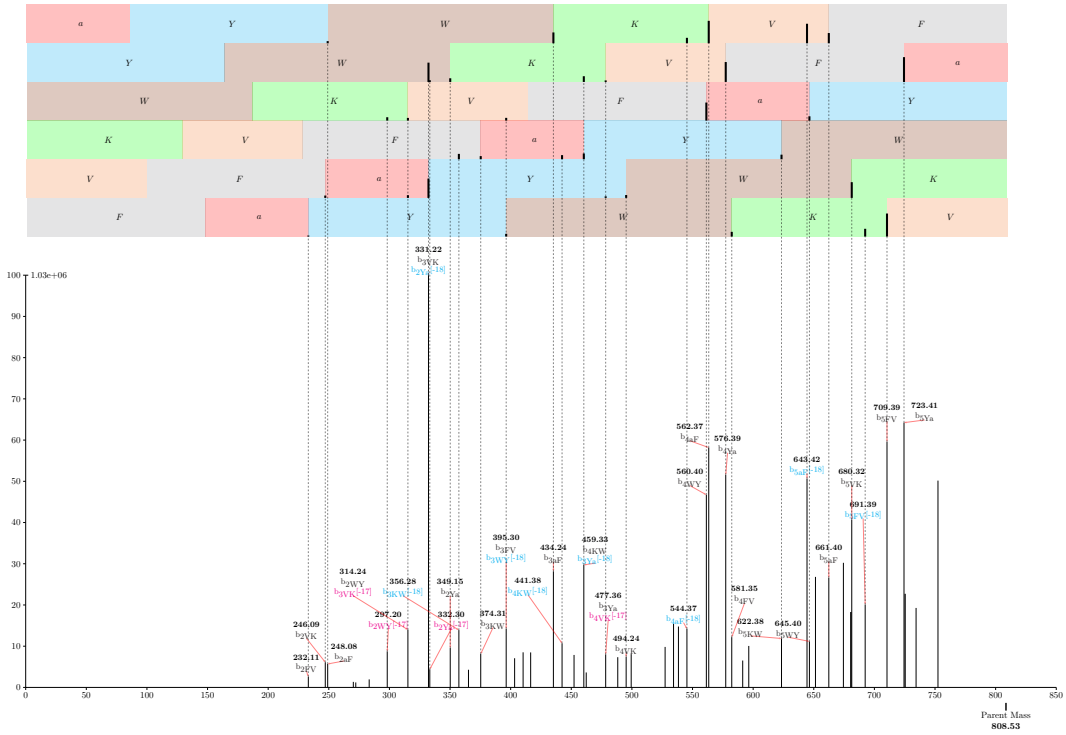


Figure 7.4: Annotated spectrum of seglitide.

7.2.2 NRP-Analysis

The input page of NRP-Analysis is very similar to the input page of NRP-Annotation. This page allows the user to run both NRP-Dereplication and NRP-Tagging. NRP-Dereplication attempts to match the spectrum against the Norine database [66], while the NRP-Tagging algorithm generates de novo reconstructions of cyclic peptides. The details of the algorithms are described elsewhere in this document and also as published work [125]. NRP-Dereplication generates an HTML table of candidate matches (Figure 7.5). The table is sortable according to the percentage of explained intensity, percentage of breaks or score of the reconstructions. Additionally, using the tooltip feature of HTML, all amino acid masses have their accurate masses displayed when the user hovers the mouse cursor over the name of the amino acid. Lastly, the user can examine the quality of the re-

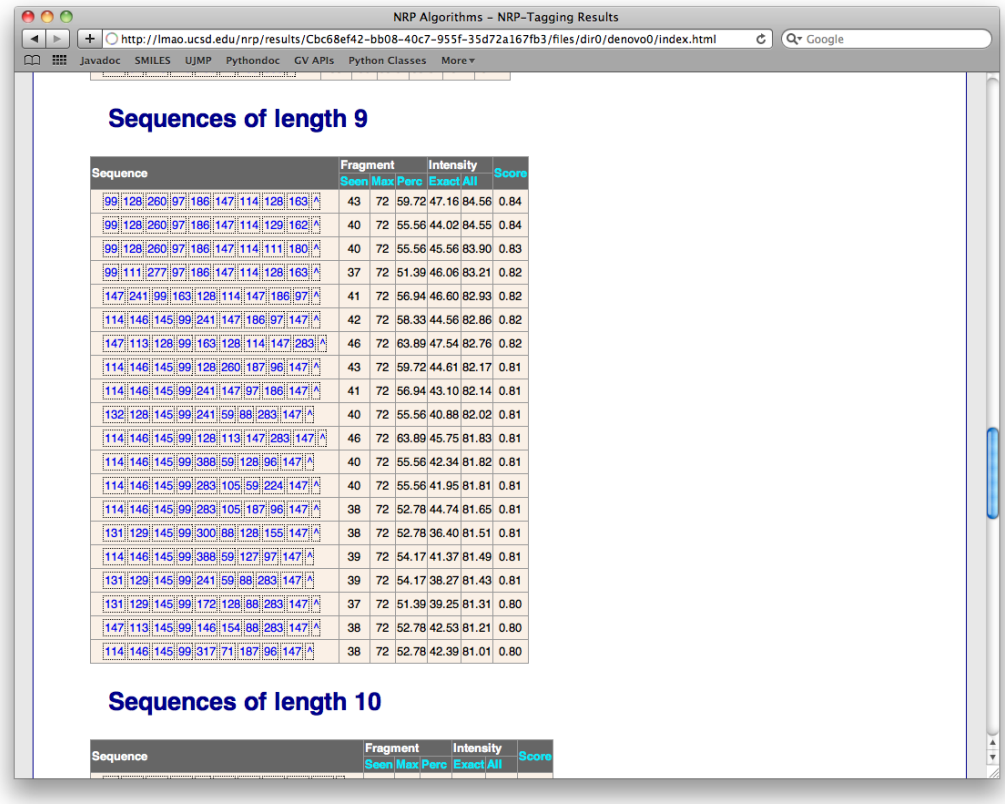
construction by following the link to the annotation page, using the dereplicated sequence and the input spectrum.

Compound Name	Amino Acid Sequence	Perc Exp Int	Perc Breaks	Dereplication Score	Annotation
tyrocidine B[+14]	[D-Phe Pro Trp D-Phe Asn Gln Tyr Val Orn[+14] Leu]	75.53	42.86	0.74	link
tyrocidine B[+14]	[D-Phe Pro Trp D-Phe Asn Gln Tyr Val Orn Leu[+14]]	74.20	41.76	0.73	link
tyrocidine B[+14]	[D-Phe[+14] Pro Trp D-Phe Asn Gln Tyr Val Orn Leu]	70.94	37.36	0.70	link
tyrocidine B[+14]	[D-Phe Pro Trp D-Phe Asn Gln Tyr Val[+14] Orn Leu]	69.33	35.16	0.68	link
tyrocidine C[-25]	[D-Phe Pro Trp D-Trp Asn Gln Tyr Val Orn[-25] Leu]	60.31	37.36	0.59	link
tyrocidine B[+14]	[D-Phe Pro Trp D-Phe Asn Gln Tyr[+14] Val Orn Leu]	59.36	28.57	0.58	link
tyrocidine C[-25]	[D-Phe Pro Trp D-Trp Asn Gln Tyr Val Orn Leu[-25]]	58.40	32.97	0.57	link
tyrocidine C[-25]	[D-Phe Pro Trp D-Trp[-25] Asn Gln Tyr Val Orn Leu]	53.63	31.87	0.52	link
tyrocidine B[+14]	[D-Phe Pro Trp D-Phe[+14] Asn Gln Tyr Val Orn Leu]	53.63	29.67	0.52	link
tyrocidine B[+14]	[D-Phe Pro[+14] Trp D-Phe Asn Gln Tyr Val Orn Leu]	51.71	36.26	0.50	link
tyrocidine C[-25]	[D-Phe[-25] Pro Trp D-Trp Asn Gln Tyr Val Orn Leu]	49.99	26.37	0.49	link

Figure 7.5: NRP-Dereplication results from the webserver.

The NRP-Tagging results are grouped by length of the reconstructed sequence. For example, Figure 7.6 shows the de novo reconstructions of length 9 for a sample run of the algorithm. The user can examine the quality of the reconstruction, and specifically the quality of each amino acid mass by clicking on the sequence inside the table. An interactive graph is presented to the user (Figure 7.7) with each peak indicating a candidate place for an amino acid mass boundary. Therefore, any distance between two peaks is a possible amino acid mass. The peak height is proportional to the confidence that there is a break at the given position, based on the input spectrum. The user can interactively display sequences using more or fewer peaks by hovering the mouse on the peaks. For example, the user can use the top 3 peaks to display a three-mass sequence, or use the top 6 peaks to display a six-mass sequence. In Figure 7.7, all 9 peaks are used to display the nine-mass sequence.

The interactivity of Figure 7.7 was achieved using only HTML, JavaScript and Cascading Style Sheets (CSS). The main advantage of using these technologies is that any modern web browser can display the figure without any plugins. As opposed to static vector graphic images generated with PyX, the HTML image displays several layers of information interactively.



Sequence	Fragment Seen	Fragment Max	Fragment Perc	Intensity Exact	Intensity All	Score
99 128 260 97 186 147 114 128 163 ^	43	72	59.72	47.16	84.56	0.84
99 128 260 97 186 147 114 129 162 ^	40	72	55.56	44.02	84.55	0.84
99 128 260 97 186 147 114 111 180 ^	40	72	55.56	45.56	83.90	0.83
99 111 277 97 186 147 114 128 163 ^	37	72	51.39	46.06	83.21	0.82
147 241 99 163 128 114 147 186 97 ^	41	72	56.94	46.60	82.93	0.82
114 146 145 99 241 147 186 97 147 ^	42	72	58.33	44.56	82.86	0.82
147 113 128 99 163 128 114 147 283 ^	46	72	63.89	47.54	82.76	0.82
114 146 145 99 128 260 187 96 147 ^	43	72	59.72	44.61	82.17	0.81
114 146 145 99 241 147 97 186 147 ^	41	72	56.94	43.10	82.14	0.81
132 128 145 99 241 59 88 283 147 ^	40	72	55.56	40.88	82.02	0.81
114 146 145 99 128 113 147 283 147 ^	46	72	63.89	45.75	81.83	0.81
114 146 145 99 388 59 128 96 147 ^	40	72	55.56	42.34	81.82	0.81
114 146 145 99 283 105 59 224 147 ^	40	72	55.56	41.95	81.81	0.81
114 146 145 99 283 105 187 96 147 ^	38	72	52.78	44.74	81.65	0.81
131 129 145 99 300 88 128 155 147 ^	38	72	52.78	36.40	81.51	0.81
114 146 145 99 388 59 127 97 147 ^	39	72	54.17	41.37	81.49	0.81
131 129 145 99 241 59 88 283 147 ^	39	72	54.17	38.27	81.43	0.81
131 129 145 99 172 128 88 283 147 ^	37	72	51.39	39.25	81.31	0.80
147 113 145 99 146 154 88 283 147 ^	38	72	52.78	42.53	81.21	0.80
114 146 145 99 317 71 187 96 147 ^	38	72	52.78	42.39	81.01	0.80

Sequence	Fragment Seen	Fragment Max	Fragment Perc	Intensity Exact	Intensity All	Score

Figure 7.6: NRP-Tagging results from the webserver.

7.3 Discussion

The work in this chapter centers on user-friendly interfaces and data representation to aid researchers make the most out of the algorithms for analysis of cyclic peptides. The use of JavaScript and CSS to generate interactive figures is a novel way to represent results that is cross-browser compatible. The other advantage of this method of data presentation is that the only requirement from the user, is a web browser, available in any computer running any operating system. The code written to generate the interactive figures can be adapted for other applications that normally generate static images allowing for better content organization with minimal software dependencies. For example, annotated mass spectra are usually static images that cannot display annotated peaks selectively (i.e. only *b*-ions or *y*-ions). As a result, the resulting image can be cluttered with text annotations when all peaks are displayed. On the other hand, annotation programs that do allow for interactivity are usually part of a proprietary software package from instrument vendors runnable only on computers that have the application installed.

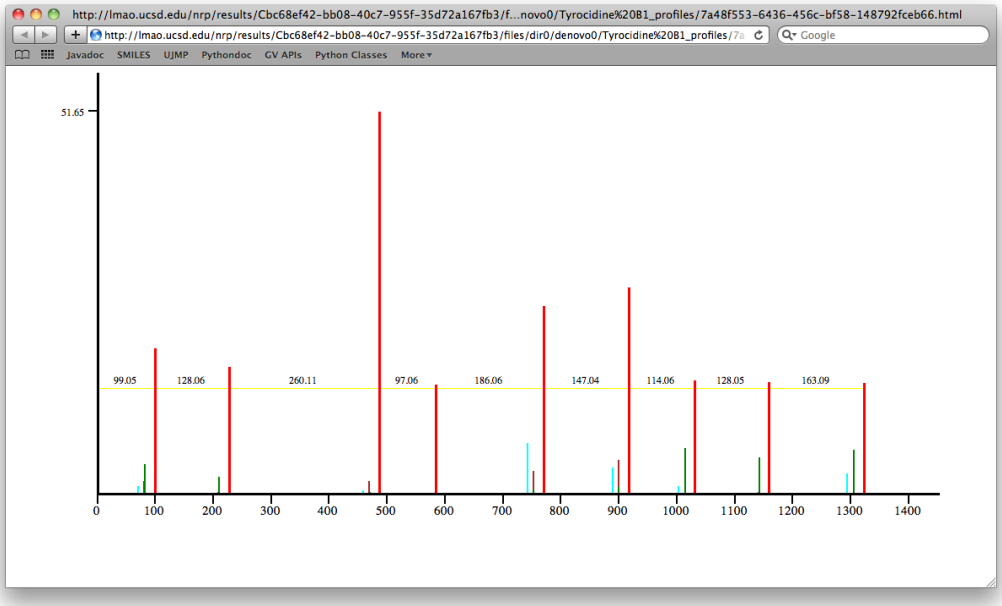


Figure 7.7: De novo sequence represented as an interactive profile.

Appendix A

Additional Tables

A.1 List of Monomers in Norine

Table A.1 lists all the monomers in the Norine database.

Table A.1: List of masses of amino acids (monomers) in NORINE sorted by mass (504 amino acids with 288 unique elemental compositions).

Mass	Name
57.02	Gly
67.04	Pyr
69.02	dh-Ala
70.01	Pya
70.04	C4:0
70.09	Put
71.04	Ala
71.04	D-Ala
71.04	NMe-Gly
71.04	bAla
72.02	D-Lac
72.02	Lac
73.05	Serol

Table A.1: List of monomers. Continued from previous page

83.04	HseL
83.04	NMe-Dha
83.04	OH-Pyr
83.04	dhAbu
84.02	C4:1(3)-OH(2)
84.06	C4:0-Me(2)
85.02	aFo-Gly
85.05	Abu
85.05	Aib
85.05	D-3OMe-Ala
85.05	D-Abu
85.05	NMe-Ala
85.05	NMe-bAla
85.09	Ivalol
85.09	Valol
86.04	C4:0-OH(3)
86.05	Dpr
87.03	D-Ser
87.03	Iser
87.03	Ser
94.04	C6:2(t2.t4)
95.04	Me-Suc
96.07	ProC
97.05	2Dh-Mabu
97.05	D-Pro
97.05	Pro
97.05	norCMA
98.07	C6:0
98.07	Me-Vaa
99.03	D-NFo-Ala

Table A.1: List of monomers. Continued from previous page

99.03	NFo-Ala
99.07	C5:0-NH2(3)
99.07	D-Ival
99.07	D-Nva
99.07	D-Val
99.07	Ival
99.07	Mab
99.07	NdMe-Ala
99.07	Nva
99.07	Val
99.10	Ileol
99.10	Leuol
100.02	C4:0-OH(2)-Ep(3)
100.05	D-Hiv
100.05	Hiv
100.06	D-Dab
100.06	Dab
100.99	dhCys
101.05	D-Hse
101.05	D-Thr
101.05	D-aThr
101.05	Hse
101.05	NMe-Ser
101.05	OH-4Abu
101.05	Thr
101.05	aThr
102.03	C4:0-OH(2.3)
103.01	Cys
103.01	D-Cys
104.03	Bz

Table A.1: List of monomers. Continued from previous page

110.03	NMe-Gly-Thz
110.08	NMe-Gln
111.03	4oxo-Pro
111.03	pGlu
111.07	3Me-Pro
111.07	4Me-Pro
111.07	5Me-Pro
111.07	CMA
111.07	Hpr
112.05	C6:0-Ep(2)
112.05	k-Leu
112.06	D-OH-cOrn
112.06	OH-cOrn
112.09	C7:0
112.09	iC7:0
113.01	Azd
113.05	3OH-Pro
113.05	4OH-Pro
113.05	D-4OH-Pro
113.05	NFo-D-Abu
113.08	D-Ile
113.08	D-Leu
113.08	D-NMe-Nva
113.08	D-NMe-Val
113.08	D-aIle
113.08	D-t-Leu
113.08	Ile
113.08	Leu
113.08	Map
113.08	NMe-Val

Table A.1: List of monomers. Continued from previous page

113.08	aIle
113.08	t-Leu
113.13	NSPD
114.03	Hap
114.03	Pda
114.04	Asn
114.04	D-Asn
114.04	NFo-Dpr
114.07	4Me-D-Hva
114.07	C6:0-OH(3)
114.07	D-Hmp
114.07	Hmp
114.08	D-Orn
114.08	Orn
115.03	Asp
115.03	D-Asp
115.03	NFo-Iser
115.06	D-bOH-Val
115.06	NMe-Thr
115.06	OMe-Thr
115.06	bOH-Val
116.05	iC5:0-OH(2.3)
116.05	iC5:0-OH(2.4)
117.02	aMe-Cys
117.04	4OH-Thr
118.03	C4:0-OH(2.3.4)
118.04	Pha
120.02	pOH-Bz
120.06	C8:3(t2.t4.t6)
121.02	Hpa

Table A.1: List of monomers. Continued from previous page

122.07	C8:0:1(7)
122.07	C8:2(2.t4)
125.05	4oxo-5Me-Pro
125.05	4oxo-Hpr
125.05	NFo-Pro
126.10	C8:0
126.10	iC8:0
127.04	bU-dAla
127.06	3OH-5Me-Pro
127.06	Ac-Aib
127.06	NFo-Val
127.10	C6:0-Me(2)-NH2(3)
127.10	D-NMe-Leu
127.10	D-NMe-aIle
127.10	Dov
127.10	Et-Nva
127.10	Hil
127.10	NMe-Ile
127.10	NMe-Leu
127.10	NMe-aIle
127.10	bMe-Ile
127.15	Spd
128.06	D-Gln
128.06	D-N2Me-Asn
128.06	Gln
128.06	N2Me-Asn
128.06	NAc-Dpr
128.06	NMe-Asn
128.06	bMe-Asn
128.08	C6:0-OMe(3)

Table A.1: List of monomers. Continued from previous page

128.08	C7:0-OH(3)
128.09	D-Lys
128.09	Lys
128.09	N-OH-Hta
128.09	bLys
129.04	Ac-Ser
129.04	D-Glu
129.04	D-bMe-Asp
129.04	Glu
129.04	bMe-Asp
129.04	bOMe-Asp
129.08	3OH-Leu
129.08	Aco
129.08	Ria
129.08	bOH-NMe-Val
129.08	gOH-NMe-Val
130.03	iC5:0-OH(2)-CA(4)
130.04	D-OH-Asn
130.04	OH-Asn
130.05	gSer
130.06	aC6:0-OH(2.3)
130.07	D-OH-Orn
130.07	OH-Orn
130.07	OH-bLys
131.02	D-OH-Asp
131.02	OH-Asp
131.04	Met
132.04	Ara
132.04	D-Ara
132.04	Lyx

Table A.1: List of monomers. Continued from previous page

132.08	Oli
133.05	D-ph-Gly
133.05	Ph-Gly
133.09	Pheol
135.01	4Cl-Thr
135.99	C4:0-OH(2.3)-Cl(4)
136.02	diOH-Bz
136.09	C8:0:1(7)-Me(2)
136.09	iC9:2(2.t4)
137.06	His
138.04	dPyr
139.06	2Me-3Me-pGlu
140.08	C6:0-Me(5.5)-oxo(2)
140.11	Argal
140.12	C9:0
140.12	aC9:0
140.12	iC9:0
141.00	MCP
141.08	4oxo-Van
141.08	Ac-Ival
141.08	Ac-Val
141.08	Ibu
141.08	NFo-Ile
141.08	NFo-Leu
141.12	NMe-OMe-Ile
141.12	NMe-bMe-Leu
141.12	NdMe-Leu
141.12	OAc-Leuol
142.07	D-bMe-Gln
142.10	C6:0-Me(2.2)-OH(3)

Table A.1: List of monomers. Continued from previous page

142.10	C7:0-Me(2)-OH(3)
142.10	C8:0-OH(3)
143.06	3Me-Glu
143.06	Aad
143.06	D-MeO-Glu
143.06	MeO-D-Glu
143.06	MeO-Glu
143.09	Act
143.09	Ere
143.09	NMe-OH-Ile
143.09	Nst
143.09	Van
144.05	N2Me-bOH-Asn
144.05	bOH-Gln
145.04	OMe-Asp
145.07	C6:0-OH(3.5)-NH2(4)
146.06	Rha
147.04	O-Met
147.05	Cl-Ile
147.07	D-Phe
147.07	D-bPhe
147.07	NMe-Ph-Gly
147.07	Phe
147.07	bPhe
148.02	Hpoe
148.05	D-Ph-Lac
148.05	Ph-Lac
149.05	D-Hpg
149.05	Hpg
150.10	iC10:2(2.t4)

Table A.1: List of monomers. Continued from previous page

150.99	CysA
150.99	D-CysA
151.04	D-F-ph-Gly
151.10	C8:0:1(7)-Me(2)-NH2(3)
152.08	C8:0:1(7)-Me(2)-OH(3)
152.12	C9:1(8)-Me(2)
153.05	OH-His
153.12	NMe-hv-Val
154.02	Ala-Thz
154.09	Cap
154.09	D-End
154.09	End
154.14	C10:0
154.14	aC10:0
154.14	iC10:0
155.09	NAc-Leu
155.09	dDap
155.13	Me-AOA
156.05	NFo-Gln
156.08	Hip
156.09	3Me-4Me-Gln
156.10	Arg
156.10	D-Arg
156.12	C8:0-Me(4)-OH(3)
156.12	C9:0-OH(3)
156.12	aC9:0-OH(3)
156.12	iC9:0-OH(3)
156.14	NtMe-Leu
157.09	Cit
157.09	D-Cit

Table A.1: List of monomers. Continued from previous page

157.11	Ist
157.11	Sta
158.07	D-Fo-OH-Orn
158.07	Fo-OH-Orn
159.05	Ahad
159.07	NMe-dPhe
160.07	2OMe-Rha
161.08	3Me-Phe
161.08	D-NMe-Phe
161.08	Hph
161.08	NMe-Phe
162.05	D-Gal
162.05	D-Glc
162.05	D-Man
162.05	Glc
162.05	bD-Gal
163.03	D-OH-dHpg
163.03	O2-Met
163.03	OH-dHpg
163.04	PT
163.06	D-Tyr
163.06	NMe-Hpg
163.06	Ph-Ser
163.06	Tyr
163.06	bTyr
164.05	4OH-D-Ph-Lac
164.06	PALOA
164.97	Cl2-Pro
165.04	Dhpg
166.09	ck-Arg

Table A.1: List of monomers. Continued from previous page

166.10	C8:0:1(7)-Me(2.2)-OH(3)
167.09	Choi
167.13	3d-NMe-Bmt
168.04	OMe-bAla-Thz
168.12	C8:1(7)-Me(2.2)-OH(3)
168.15	aC11:0
169.11	Bmt
169.11	Dap
169.17	GSpd
170.08	5OH-Cap
170.09	Hysp
170.12	Har
170.13	C10:0-OH(3)
170.13	C8:0-Me(2.2)-OH(3)
170.13	C9:0-Me(2)-OH(3)
170.13	iC8:0-Me(2.4)-OH(3)
171.09	NOMe-Ac-Val
171.13	aC9:0-OH(2)-NH2(3)
171.13	dDil
172.08	Ac-OH-Orn
172.08	D-Ac-OH-Orn
172.10	Trpol
174.96	MdCP
175.07	bbMe2-O-Met
175.10	Apv
177.08	Hty
177.08	NMe-Tyr
177.08	bOH-NMe-Phe
178.07	PAOA
178.14	C10:0:1(9)-Me(2.4)

Table A.1: List of monomers. Continued from previous page

178.14	iC12:2(2.t4)
179.06	bOH-Tyr
179.06	diOH-Phe
180.04	Pro-Thz
180.15	C12:1(5)
181.07	Aca
182.08	D-Har
182.12	v-Arg
182.17	C12:0
182.17	iC12:0
183.01	Cl-Hpg
183.13	NMe-Bmt
184.06	dh-Trp
184.08	U4oxo-Van
184.10	k-Arg
184.15	C10:0-Me(2)-OH(3)
184.15	C10:0-Me(4)-OH(3)
184.15	C11:0-OH(3)
184.15	iC11:0-OH(3)
185.08	Dpy
185.14	Dil
186.06	Doe
186.08	D-Trp
186.08	Trp
186.11	hk-Arg
187.06	dv-Tyr
188.09	diOH-Arg
189.08	Ac-Phe
189.08	v-Tyr
190.07	D-Kyn

Table A.1: List of monomers. Continued from previous page

190.07	Kyn
190.10	CFA
190.11	NMe-MeA-Phe
190.11	NMe-OMe-TyrC
191.09	3Me-Hty
191.09	Ahv
191.09	D-NMe-OMe-Tyr
191.09	NMe-Hty
191.09	NMe-OMe-Tyr
191.09	e-Tyr
192.15	aC13:2(2.t4)
193.01	Cl2-NMe-dhLeu
193.07	bOMe-Tyr
194.02	OSu-Hmp
194.14	MeOx-Ile
194.17	aC13:1(3)
194.17	iC13:1(3)
195.02	Cl2-NMe-Leu
196.15	C10:0-Me(2.4)-oxo(9)
196.15	C12:1(5)-OH(3)
196.15	C9:1(4)-Me(2.4.6)-OH(8)
196.18	aC13:0
197.02	Cl-Tyr
197.11	C10:0-NH2(2)-Ep(9)-oxo(8)
197.14	Me2-Bmt
198.16	C11:0-Me(2)-OH(3)
198.16	C12:0-OH(3)
198.16	iC12:0-OH(3)
200.08	NAc-Fo-OH-Orn
200.09	1Me-Trp

Table A.1: List of monomers. Continued from previous page

202.07	OH-Trp
202.07	pTrp
204.13	NMe-Me2A-Phe
205.07	v-OH-Tyr
205.11	Amv
206.04	NMe-Lan
208.18	C14:1(7)
208.18	iC14:1(3)
210.00	D-PO-Asn
210.05	PMST
210.13	C9:1(Me4)-Me(2.4.6)-OH(8)-Oxo(5)
210.20	C14:0
211.04	Cl-NMe-Tyr
211.04	D-Cl-NMe-Tyr
211.19	C13:0-NH2(3)
212.18	C13:0-OH(3)
212.18	aC13:0-OH(3)
212.18	iC13:0-OH(3)
213.02	bOH-Cl-Tyr
214.10	Ahp
216.09	NMe-OH-Trp
216.11	Daz
216.12	Agdha
216.97	Cl2-Hpg
219.05	DMOG
219.09	NOMe-Ac-D-Phe
220.04	D-Cl-Trp
221.01	DHPT
222.20	aC15:1(3)
224.14	C13:2(t4.t6)-OH(2.3)

Table A.1: List of monomers. Continued from previous page

224.14	aC11:2(4.6)-Me(2.6)-OH(2.3)
224.98	Br-Phe
225.21	C14:0-NH2(3)
225.21	iC14:0-NH2(3)
226.19	C14:0-OH(3)
226.19	iC14:0-OH(3)
226.97	Cl3-NMe-dhLeu
228.09	Ac-Trp
228.13	bbMe-NMe-Trp
228.17	C10:0-Me(2.2.4)-OH(3.7)
228.98	Cl3-NMe-Leu
230.05	Phe-Thz
230.07	D-COOH-Trp
234.06	NMe-Cl-Trp
236.21	C16:1(7)
236.21	C16:1(9)
238.16	C11:2(t2.t8)-Me(2.6.8)-OH(5.7)
238.16	C14:2(t4.t6)-OH(2.3)
238.23	C16:0
238.99	bMe-Br-Phe
239.22	C15:0-NH2(3)
239.22	aC15:0-NH2(3)
239.22	iC15:0-NH2(3)
240.21	C15:0-OH(3)
240.21	aC15:0-OH(3)
240.21	iC15:0-OH(3)
240.97	Br-Tyr
240.97	bOH-Br-Phe
242.14	bbNMe-NMe-Trp
242.19	C10:0-Me(2.2.4)-OH(3)-OMe(7)

Table A.1: List of monomers. Continued from previous page

242.19	C14:0-OH(3.4)
243.13	C8:2(5.7)-Me(6)-OH(4)-NH2(3)-Ph(8)
244.98	Cl3-2OH-NMe-Leu
244.98	Cl3-5OH-NMe-Leu
250.05	NMe-Cl-OH-Trp
252.17	C15:2(t4.t6)-OH(2.3)
252.21	C16:1(9)-OH(3)
253.24	C16:0-NH2(3)
253.24	iC16:0-NH2(3)
254.22	C16:0-OH(3)
254.22	iC16:0-OH(3)
254.99	D-Br-NMe-Tyr
258.10	N1-COOH-bhTrp
259.10	ChrI
259.10	ChrP
261.11	ChrD
263.05	D-Cl-CONH2-Trp
263.12	C8:1(7)-OH(2.4.5)-NH2(3)-Ph(8)
263.99	Br-Trp
264.25	C18:1(9)
266.19	DHMDA
267.26	aC17:0-NH2(3)
267.26	iC17:0-NH2(3)
268.24	aC17:0-OH(3)
268.24	iC17:0-OH(3)
270.22	C16:0-OH(3.4)
272.06	PTTA
278.01	NMe-Br-Trp
279.98	Br-OH-Trp
280.24	C18:1(9)-OH(3)

Table A.1: List of monomers. Continued from previous page

285.07	ChrA
299.19	DMAdda
302.98	D-I-NMe-Tyr
310.06	ChrAct
313.20	Adda
321.04	C8:2(5.7)-Me(6)-OH(4)-NH2(3)-brPh(8)
329.16	C12:3(7.9.11)-Me(6)-OH(2.4.5)-NH2(3)-Ph(12)
333.16	C10:2(7.9)-OH(2.4.5)-NH2(3)-ePh(10)
341.20	ADMAdda
343.18	C12:3(7.9.11)-Me(6.10)-OH(2.4.5)-NH2(3)-Ph(12)
363.20	C12:1(11)-Me(6)-OH(2.4.5)-NH2(3)-mPhe(11)

Bibliography

[1] J. Eng, A. McCormack, and J. Yates, "An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database," *Journal of the American Society for Mass Spectrometry*, vol. 5, pp. 976–989, November 1994.

[2] D. N. Perkins, D. J. C. Pappin, D. M. Creasy, and J. S. Cottrell, "Probability-based protein identification by searching sequence databases using mass spectrometry data," *Electrophoresis*, vol. 20, no. 18, pp. 3551–3567, 1999.

[3] S. Tanner, H. Shu, A. Frank, L. C. Wang, E. Zandi, M. Mumby, P. A. Pevzner, and V. Bafna, "Inspect: Identification of posttranslationally modified peptides from tandem mass spectra," *Analytical Chemistry*, vol. 77, no. 14, pp. 4626–4639, 2005.

[4] D. L. Tabb, M. J. MacCoss, C. C. Wu, S. D. Anderson, and J. R. Yates, "Similarity among tandem mass spectra from proteomic experiments: Detection, significance, and utility," *Analytical Chemistry*, vol. 75, pp. 2470–2477, 04 2003.

[5] I. Beer, E. Barnea, T. Ziv, and A. Admon, "Improving large-scale proteomics by clustering of mass spectrometry data," *PROTEOMICS*, vol. 4, pp. 950–960, April 2004.

[6] D. L. Tabb, M. R. Thompson, G. Khalsa-Moyers, N. C. VerBerkmoes, and W. H. McDonald, "Ms2grouper: Group assessment and synthetic replacement of duplicate proteomic tandem mass spectra," *Journal of the American Society for Mass Spectrometry*, vol. 16, pp. 1250–1261, 8 2005.

[7] A. M. Frank, N. Bandeira, Z. Shen, S. Tanner, S. P. Briggs, R. D. Smith, and P. A. Pevzner, "Clustering millions of tandem mass spectra," *Journal of Proteome Research*, vol. 7, pp. 113–122, 12 2007.

[8] K. Jeong, N. Bandeira, S. Kim, and P. A. Pevzner, "Gapped spectral dictionaries and their applications for database searches of tandem mass spectra," *Mol Cell Proteomics*, vol. in press, 2010.

[9] S. Kim, N. Bandeira, and P. A. Pevzner, "Spectral profiles: A novel representation of tandem mass spectra and its applications for de novo peptide sequencing and identification," *Mol Cell Proteomics*, pp. M800535–MCP200, 2009.

[10] S. Kim, N. Gupta, N. Bandeira, and P. A. Pevzner, "Spectral dictionaries," *Mol Cell Proteomics*, vol. 8, no. 1, pp. 53–69, 2009.

[11] J. W. Gray and C. Collins, "Genome changes and gene expression in human solid tumors," *Carcinogenesis*, vol. 21, no. 3, pp. 443–452, 2000.

[12] M. Ehrlich, *DNA Alterations in Cancer: Genetic and Epigenetic Changes*. Eaton Pub, 2000.

[13] S. Volik, S. Zhao, K. Chin, J. H. Brebner, D. R. Herndon, Q. Tao, D. Kowbel, G. Huang, A. Lapuk, W. L. Kuo, G. Magrane, P. J. de Jong, J. W. Gray, and C. Collins, "End-sequence profiling: Sequence-based analysis of aberrant genomes," *Proceedings of the National Academy of Sciences*, vol. 100, no. 13, pp. 7696–7701, 2003.

[14] S. E. Artandi, S. Chang, S.-L. Lee, S. Alson, G. J. Gottlieb, L. Chin, and R. A. DePinho, "Telomere dysfunction promotes non-reciprocal translocations and epithelial cancers in mice," *Nature*, vol. 406, pp. 641–645, 08 2000.

[15] F. Mitelman, B. Johansson, and F. Mertens, "Fusion genes and rearranged genes as a linear function of chromosome aberrations in cancer," *Nature Genetics*, vol. 36, no. 4, pp. 331–334, 2004.

[16] S. Volik, B. J. Raphael, G. Huang, M. R. Stratton, G. Bignel, J. Murnane, J. H. Brebner, K. Bajsarowicz, P. L. Paris, Q. Tao, D. Kowbel, A. Lapuk, D. A. Shagin, I. A. Shagina, J. W. Gray, J. F. Cheng, P. J. de Jong, P. A. Pevzner, and C. Collins, "Decoding the fine-scale structure of a breast cancer genome and transcriptome," *Genome Res.*, vol. 16, no. 3, pp. 394–404, 2006.

[17] B. J. Raphael, S. Volik, C. Collins, and P. A. Pevzner, "Reconstructing tumor genome architectures," *Bioinformatics*, vol. 19, pp. 162–171, 2003.

[18] B. J. Raphael and P. A. Pevzner, "Reconstructing tumor amplisomes," *Bioinformatics*, vol. 20, no. Suppl 1, pp. i265–273, 2004.

[19] P. Ng, C. L. Wei, W. K. Sung, K. P. Chiu, L. Lipovich, C. C. Ang, S. Gupta, A. Shahab, A. Ridwan, C. H. Wong, E. T. Liu, and Y. Ruan, "Gene identification signature (GIS) analysis for transcriptome characterization and genome annotation," *Nature Methods*, vol. 2, no. 2, pp. 105–111, 2005.

[20] Y. Ruan, H. S. Ooi, S. W. Choo, K. P. Chiu, X. D. Zhao, K. Srinivasan, F. Yao, C. Y. Choo, J. Liu, P. Ariyaratne, W. G. Bin, V. A. Kuznetsov, A. Shahab, W.-K. Sung, G. Bourque, N. Palanisamy, and C.-L. Wei, "Fusion transcripts and transcribed retrotransposed loci discovered through comprehensive transcriptome analysis using Paired-End diTags (PETs)," *Genome Res.*, vol. 17, no. 6, pp. 828–838, 2007.

[21] R. Kurzrock, H. M. Kantarjian, B. J. Druker, and M. Talpaz, "Philadelphia chromosome-positive leukemias: From basic mechanisms to molecular therapeutics," *Ann Intern Med*, vol. 138, no. 10, pp. 819–830, 2003.

[22] P. A. Futreal, A. Kasprzyk, E. Birney, J. C. Mullikin, R. Wooster, and M. R. Stratton, "Cancer and genomics," *Nature*, vol. 409, no. 6822, pp. 850–852, 2001.

[23] Y. Hahn, T. K. Bera, K. Gehlhaus, I. R. Kirsch, I. H. Pastan, and B. Lee, "Finding fusion genes resulting from chromosome rearrangement by analyzing the expressed sequence databases," *Proceedings of the National Academy of Sciences*, vol. 101, no. 36, pp. 13257–13261, 2004.

[24] K. S. J. Elenitoba-Johnson, D. K. Crockett, J. A. Schumacher, S. D. Jenson, C. M. Coffin, A. L. Rockwood, and M. S. Lim, "Proteomic identification of oncogenic chromosomal translocation partners encoding chimeric anaplastic lymphoma kinase fusion proteins," *Proceedings of the National Academy of Sciences*, vol. 103, no. 19, pp. 7402–7407, 2006.

[25] C. Touriol, C. Greenland, L. Lamant, K. Pulford, F. Bernard, T. Rousset, D. Y. Mason, and G. Delsol, "Further demonstration of the diversity of chromosomal changes involving 2p23 in ALK-positive lymphoma: 2 cases expressing ALK kinase fused to CLTCL (clathrin chain polypeptide-like)," *Blood*, vol. 95, no. 10, pp. 3204–3207, 2000.

[26] L. Lamant, N. Dastugue, K. Pulford, G. Delsol, and B. Mariame, "A new fusion gene TPM3-ALK in anaplastic large cell lymphoma created by a (1;2)(q25;p23) translocation," *Blood*, vol. 93, no. 9, pp. 3088–3095, 1999.

[27] L. Hernandez, M. Pinyol, S. Hernandez, S. Bea, K. Pulford, A. Rosenwald, L. Lamant, B. Falini, G. Ott, D. Y. Mason, G. Delsol, and E. Campo, "TRK-fused gene (TFG) is a new partner of ALK in anaplastic large cell lymphoma producing two structurally different TFG-ALK translocations," *Blood*, vol. 94, no. 9, pp. 3265–3268, 1999.

[28] M. Trinei, L. Lanfrancone, E. Campo, K. Pulford, D. Y. Mason, P.-G. Pelicci, and B. Falini, "A new variant anaplastic lymphoma kinase (ALK)-fusion protein (ATIC-ALK) in a case of ALK-positive anaplastic large cell lymphoma," *Cancer Res*, vol. 60, no. 4, pp. 793–798, 2000.

[29] F. Tort, M. Pinyol, K. Pulford, G. Roncador, L. Hernandez, I. Nayach, H. C. Kluin-Nelemans, P. Kluin, C. Touriol, G. Delsol, D. Mason, and E. Campo, "Molecular characterization of a new ALK translocation involving moesin (MSN-ALK) in anaplastic large cell lymphoma," *Lab Invest*, vol. 81, no. 3, pp. 419–426, 2001.

[30] S. Tanner, Z. Shen, J. Ng, L. Florea, R. Guigo, S. P. Briggs, and V. Bafna, "Improving gene annotation using peptide mass spectrometry," *Genome Research*, vol. 17, no. 2, pp. 231–239, 2007.

[31] V. Dancik, T. Addona, K. Clauser, J. Vath, and P. Pevzner, "De novo peptide sequencing via tandem mass spectrometry," *J Comput Biol*, vol. 6, pp. 327–342, 1999.

[32] D. Tsur, S. Tanner, E. Zandi, V. Bafna, and P. Pevzner, "Identification of post-translational modifications by blind search of mass spectra," *Nature Biotechnology*, vol. 23, no. 12, pp. 1562–1567, 2005.

[33] V. Bafna and N. Edwards, "On de novo interpretation of tandem mass spectra for peptide identification," *Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB 2003)*, pp. 9–18, 2003.

[34] G. S. Omenn, "The human proteome organization plasma proteome project pilot phase: Reference specimens, technology platform comparisons, and standardized data submissions and analyses," *Journal of Proteomics*, vol. 4, no. 5, pp. 1235–1240 1235–1240 1235–1240 1235–1240 1235–1240 1235–1240, 2004.

[35] G. S. Omenn, D. J. States, M. Adamski, T. W. Blackwell, R. Menon, H. Hermjakob, R. Apweiler, B. B. Haab, R. J. Simpson, J. S. Eddes, E. A. Kapp, R. L. Moritz, D. W. Chan, A. J. Rai, A. Admon, R. Aebersold, J. Eng, W. S. Hancock, S. A. Hefta, H. Meyer, Y.-K. Paik, J.-S. Yoo, P. Ping, J. Pounds, J. Adkins, X. Qian, R. Wang, V. Wasinger, C. Y. Wu, X. Zhao, R. Zeng, A. Archakov, A. Tsugita, I. Beer, A. Pandey, M. Pisano, P. Andrews, H. Tammen, D. W. Speicher, S. M. Hanash, G. S. Omenn, D. J. States, M. Adamski, T. W. Blackwell, R. Menon, H. Hermjakob, R. Apweiler, B. B. Haab, R. J. Simpson, J. S. Eddes, E. A. Kapp, R. L. Moritz, D. W. Chan, A. J. Rai, A. Admon, R. Aebersold, J. Eng, W. S. Hancock, S. A. Hefta, H. Meyer, Y.-K. Paik, J.-S. Yoo, P. Ping, J. Pounds, J. Adkins, X. Qian, R. Wang, V. Wasinger, C. Y. Wu, X. Zhao, R. Zeng, A. Archakov, A. Tsugita, I. Beer, A. Pandey, M. Pisano, P. Andrews, H. Tammen, D. W. Speicher, and S. M. Hanash, "Overview of the hupo plasma proteome project: Results from the pilot phase with 35 collaborating laboratories and multiple

analytical groups, generating a core dataset of 3020 proteins and a publicly-available database," *J Proteomics*, vol. 5, no. 13, pp. 3226–3245, 2005.

[36] D. Gusfield, *Algorithms on strings, trees, and sequences: computer science and computational biology.* New York, NY, USA: Cambridge University Press, 1997.

[37] C. Y. Park, A. A. Klammer, L. Käll, M. J. MacCoss, and W. S. Noble, "Rapid and accurate peptide identification from tandem mass spectra," *Journal of Proteome Research*, vol. 7, no. 7, pp. 3022–3027, 2008.

[38] B. Ma, K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby, and G. Lajoie, "PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry," *Rapid Communications in Mass Spectrometry*, vol. 17, no. 20, pp. 2337–2342, 2003.

[39] A. M. Frank and P. A. Pevzner, "PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling," *Anal. Chem.*, vol. 77, pp. 964–973, 2005.

[40] I. V. Shilov, S. L. Seymour, A. A. Patel, A. Loboda, W. H. Tang, S. P. Keating, C. L. Hunter, L. M. Nuwaysir, and D. A. Schaeffer, "The paragon algorithm, a next generation search engine that uses sequence temperature values and feature probabilities to identify peptides from tandem mass spectra," *Molecular & Cellular Proteomics*, vol. 6, no. 9, pp. 1638–1655, 2007.

[41] S. Kim, N. Gupta, and P. A. Pevzner, "Spectral probabilities and generating functions of tandem mass spectra: A strike against decoy databases," *Journal of Proteome Research*, vol. 7, pp. 3354–3363, 07 2008.

[42] J. Ng and P. A. Pevzner, "Algorithm for identification of fusion proteins via mass spectrometry," *Journal of Proteome Research*, vol. 7, no. 1, pp. 89–95, 2008.

[43] N. E. Castellana, S. H. Payne, Z. Shen, M. Stanke, V. Bafna, and S. P. Briggs, "Discovery and revision of Arabidopsis genes by proteogenomics," *Proceedings of the National Academy of Sciences*, vol. 105, no. 52, pp. 21034–21038, 2008.

[44] J. D. Jaffe, N. Stange-Thomann, C. Smith, D. DeCaprio, S. Fisher, J. Butler, S. Calvo, T. Elkins, M. G. FitzGerald, N. Hafez, C. D. Kodira, J. Major, S. Wang, J. Wilkinson, R. Nicol, C. Nusbaum, B. Birren, H. C. Berg, and G. M. Church, "The complete genome and proteome of mycoplasma mobile," *Genome Research*, vol. 14, no. 8, pp. 1447–1461, 2004.

[45] N. Gupta, S. Tanner, N. Jaitly, J. Adkins, M. Lipton, R. Edwards, M. Romine, A. Osterman, V. Bafna, R. Smith, and P. Pevzner, "Whole

proteome analysis of post-translational modifications: applications of mass-spectrometry for proteogenomic annotation," *Genome Res.*, vol. 17, pp. 1362–1377, 2007.

[46] G. E. Merrihew, C. Davis, B. Ewing, G. Williams, L. Käll, B. E. Frewen, W. S. Noble, P. Green, J. H. Thomas, and M. J. MacCoss, "Use of shotgun proteomics for the identification, confirmation, and correction of c. elegans gene annotations," *Genome Research*, vol. 18, no. 10, pp. 1660–1669, 2008.

[47] R. D. Knight, S. J. Freeland, and L. F. Landweber, "Rewiring the keyboard: evolvability of the genetic code," *Nat Rev Genet*, vol. 2, pp. 49–58, 01 2001.

[48] F. Abascal, D. Posada, R. D. Knight, and R. Zardoya, "Parallel evolution of the genetic code in arthropod mitochondrial genomes," *PLoS Biol*, vol. 4, p. e127, 04 2006.

[49] P. Nielsen and A. Krogh, "Large-scale prokaryotic gene prediction and comparison to genome annotation," *Bioinformatics*, vol. 21, no. 24, pp. 4322–4329, 2005.

[50] J. Besemer, A. Lomsadze, and M. Borodovsky, "Genemarks: a self-training method for prediction of gene starts in microbial genomes. implications for finding sequence motifs in regulatory regions," *Nucleic Acids Research*, vol. 29, no. 12, pp. 2607–2618, 2001.

[51] N. Gupta and P. A. Pevzner, "False discovery rates of protein identifications: A strike against the two-peptide rule," *Journal of Proteome Research*, vol. 8, pp. 4173–4181, 07 2009.

[52] S. Kim, N. Mischerikow, N. Bandeira, J. D. Navarro, L. Wich, S. Mohammed, A. J. R. Heck, and P. A. Pevzner, "The generating function of cid, etd, and cid/etd pairs of tandem mass spectra: Applications to database search," *Mol Cell Proteomics*, vol. 9, no. 12, pp. 2840–2852, 2010.

[53] D. M. Creasy and J. S. Cottrell, "Unimod: Protein modifications for mass spectrometry," *PROTEOMICS*, vol. 4, no. 6, pp. 1534–1536, 2004.

[54] S. A. Sieber and M. A. Marahiel, "Molecular Mechanisms Underlying Nonribosomal Peptide Synthesis: Approaches to New Antibiotics," *Chem Rev*, vol. 105, no. 2, pp. 715–38, 2005.

[55] D. J. Newman and G. M. Cragg, "Natural products as sources of new drugs over the last 25 years," *Journal of Natural Products*, vol. 70, no. 3, pp. 461–477, 2007.

[56] H. Luesch, P. Williams, W. Yoshida, R. Moore, and V. Paul, "Ulongamides A-F, New Beta-Amino Acid-Containing Cyclodepsipeptides from Palauan Collections of the Marine Cyanobacterium *Lyngbya sp.*," *Journal of Natural Products*, vol. 65, no. 7, pp. 996–1000, 2002.

[57] T. Hamada, S. Matsunaga, G. Yano, and N. Fusetani, "Polytheonamides A and B, Highly Cytotoxic, Linear Polypeptides with Unprecedented Structural Features, from the Marine Sponge, *Theonella swinhoei*," *J Am Chem Soc*, vol. 127, no. 1, pp. 110–8, 2005.

[58] C. M. Ireland, A. R. Durso, R. A. Newman, and M. P. Hacker, "Antineoplastic Cyclic Peptides from the Marine Tunicate *Lissoclinum patella*," *J. Org. Chem*, vol. 47, pp. 360–361, 1982.

[59] J. Li, A. Burgett, L. Esser, C. Amezcua, and P. Harran, "Total synthesis of nominal diazonamides: Part 2. on the true structure and origin of natural isolates," *Angew. Chem Intl. Ed. Engl.*, vol. 40, pp. 4770–4773, 2001.

[60] G. Lang, N. A. Mayhudin, M. I. Mitova, L. Sun, S. van der Sar, J. W. Blunt, A. L. J. Cole, G. Ellis, H. Laatsch, and M. H. G. Munro, "Evolving trends in the dereplication of natural product extracts: New methodology for rapid, small-scale investigation of natural product extracts," *Journal of Natural Products*, vol. 71, no. 9, pp. 1595–1599, 2008.

[61] T. Krishnamurthy, L. Szafraniec, D. F. Hunt, J. Shabanowitz, J. R. Yates, C. R. Hauer, W. W. Carmichael, O. Skulberg, G. A. Codd, and S. Missler, "Structural characterization of toxic cyclic peptides from blue-green algae by tandem mass spectrometry," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 86, no. 3, pp. 770–774, 1989.

[62] W. H. Gerwick, Z. D. Jiang, S. K. Agarwal, and B. T. Farmer, "Total structure of hormothamnin a, a toxic cyclic undecapeptide from the tropical marine cyanobacterium hormothamnion enteromorphoides," *Tetrahedron*, vol. 48, no. 12, pp. 2313 – 2324, 1992.

[63] M. Barber, D. J. Bell, M. R. Morris, L. W. Tetler, J. J. Monaghan, W. E. Morden, B. W. Bycroft, and B. N. Green, "An Investigation of the Tyrothricin Complex by Tandem Mass Spectrometry," *International Journal of Mass Spectrometry and Ion Processes*, vol. 122, pp. 143–151, 1992.

[64] G. Hitzeroth, J. Vater, P. Franke, K. Gebhardt, and H.-P. Fiedler, "Whole Cell Matrix-Assisted Laser Desorption/Ionization Time-of-Flight Mass Spectrometry and in situ Structure Analysis of Streptocidins, a Family of Tyrocidine-like Cyclic Peptides," *Rapid Communications in Mass Spectrometry*, vol. 19, no. 20, pp. 2935–2942, 2005.

[65] M. Welker, B. Marsálek, L. Sejnohová, and H. von Döhren, "Detection and identification of oligopeptides in microcystis (cyanobacteria) colonies: Toward an understanding of metabolic diversity," *Peptides*, vol. 27, no. 9, pp. 2090 – 2103, 2006.

[66] S. Caboche, M. Pupin, V. Leclere, A. Fontaine, P. Jacques, and G. Kucherov, "NORINE: a database of nonribosomal peptides," *Nucl. Acids Res.*, vol. 36, no. suppl 1, pp. D326–331, 2008.

[67] S. S. Skiena and G. Sundaram, "A Partial Digest Approach to Restriction Site Mapping," *Bulletin of Mathematical Biology*, vol. 56, no. 2, pp. 275–94, 1994.

[68] J. Rosenblatt and P. D. Seymour, "The Structure of Homometric Sets," *SIAM Journal on Algebraic and Discrete Methods*, vol. 3, no. 3, pp. 343–350, 1982.

[69] P. A. Pevzner, V. Dancik, and C. Tang, "Mutation-Tolerant Protein Identification by Mass Spectrometry," *J Comput Biol*, vol. 7, no. 6, pp. 777–787, 2000.

[70] A. R. Dongre, J. L. Jones, A. Somogyi, and V. H. Wysocki, "Influence of Peptide Composition, Gas-Phase Basicity, and Chemical Modification on Fragmentation Efficiency: Evidence for the Mobile Proton Model," *Journal of the American Chemical Society*, vol. 118, no. 35, pp. 8365–8374, 1996.

[71] V. H. Wysocki, G. Tsaprailis, L. L. Smith, and L. A. Breci, "Mobile and localized protons: a framework for understanding peptide dissociation," *Journal of Mass Spectrometry*, vol. 35, no. 12, pp. 1399–1406, 2000.

[72] Z. Zhang, "Prediction of low-energy collision-induced dissociation spectra of peptides," *Analytical Chemistry*, vol. 76, no. 14, pp. 3908–3922, 2004.

[73] A. M. Frank, "Predicting intensity ranks of peptide fragment ions," *Journal of Proteome Research*, vol. 8, no. 5, pp. 2226–2240, 2009.

[74] L. Mo, D. Dutta, Y. Wan, and T. Chen, "MSNovo: A Dynamic Programming Algorithm for De Novo Peptide Sequencing via Tandem Mass Spectrometry," *Anal Chem*, vol. 79, pp. 4870–4878, 2007.

[75] A. W. Schultz, D.-C. Oh, J. R. Carney, R. T. Williamson, D. W. Udwary, P. R. Jensen, S. J. Gould, W. Fenical, and B. S. Moore, "Biosynthesis and structures of cyclomarins and cyclomarazines, prenylated cyclic peptides of marine actinobacterial origin," *Journal of the American Chemical Society*, vol. 130, no. 13, pp. 4507–4516, 2008.

[76] E. W. Schmidt, J. T. Nelson, D. A. Rasko, S. Sudek, J. A. Eisen, M. G. Haygood, and J. Ravel, "Patellamide a and c biosynthesis by a microcin-like pathway in prochloron didemni, the cyanobacterial symbiont of lissoclinum patella," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 20, pp. 7315–7320, 2005.

[77] A. B. Pomilio, M. E. Battista, and A. A. Vitale, "Naturally-occurring cyclopeptides: Structures and bioactivity," *Current Organic Chemistry*, vol. 10, pp. 2075–2121(47), 2006.

[78] K. L. Rinehart, K. Harada, M. Namikoshi, C. Chen, C. A. Harvis, M. H. G. Munro, J. W. Blunt, P. E. Mulligan, V. R. Beasley, and . et al., "Nodularin, microcystin, and the configuration of adda," *Journal of the American Chemical Society*, vol. 110, pp. 8557–8558, 12 1988.

[79] N. Gupta, S. C. Pant, R. Vijayaraghavan, and P. V. L. Rao, "Comparative toxicity evaluation of cyanobacterial cyclic peptide toxin microcystin variants (lr, rr, yr) in mice," *Toxicology*, vol. 188, pp. 285–296, 6 2003.

[80] M. T. Holden, S. Ram Chhabra, R. De Nys, P. Stead, N. J. Bainton, P. J. Hill, M. Manefield, N. Kumar, M. Labatte, D. England, S. Rice, M. Givskov, G. P. Salmond, G. S. Stewart, B. W. Bycroft, S. Kjelleberg, and P. Williams, "Quorum-sensing cross talk: isolation and chemical characterization of cyclic dipeptides from pseudomonas aeruginosa and other gram-negative bacteria," *Molecular Microbiology*, vol. 33, no. 6, pp. 1254–1266, 1999.

[81] M. Ibrahim, A. Guillot, F. Wessner, F. Algaron, C. Besset, P. Courtin, R. Gardan, and V. Monnet, "Control of the transcription of a short gene encoding a cyclic peptide in streptococcus thermophilus: a new quorum-sensing system?," *The Journal of Bacteriology*, vol. 189, pp. 8844–8854, 12 2007.

[82] O. Poupel and I. Tardieux, "Toxoplasma gondii motility and host cell invasiveness are drastically impaired by jasplakinolide, a cyclic peptide stabilizing f-actin," *Microbes and Infection*, vol. 1, pp. 653–662, 7 1999.

[83] S. S. Branda, F. Chu, D. B. Kearns, R. Losick, and R. Kolter, "A major protein component of the bacillus subtilis biofilm matrix," *Molecular Microbiology*, vol. 59, no. 4, pp. 1229–1238, 2006.

[84] P. D. Straight, J. M. Willey, and R. Kolter, "Interactions between streptomyces coelicolor and bacillus subtilis: Role of surfactants in raising aerial structures," *The Journal of Bacteriology*, vol. 188, pp. 4918–4925, 7 2006.

[85] M. H. J. Sturme, J. Nakayama, D. Molenaar, Y. Murakami, R. Kunugi, T. Fujii, E. E. Vaughan, M. Kleerebezem, and W. M. de Vos, "An agr-like two-component regulatory system in Lactobacillus plantarum is involved in production of a novel cyclic peptide and regulation of adherence," *The Journal of Bacteriology*, vol. 187, pp. 5224–5235, 8 2005.

[86] A. Jegorov, M. Hajduch, M. Sulc, and V. Havlicek, "Nonribosomal cyclic peptides: specific markers of fungal infections," *Journal of Mass Spectrometry*, vol. 41, no. 5, pp. 563–576, 2006.

[87] J. L. Italia, V. Bhardwaj, and M. N. V. Ravi Kumar, "Disease, destination, dose and delivery aspects of ciclosporin: the state of the art," *Drug Discovery Today*, vol. 11, pp. 846–854, 9 2006.

[88] J. Hannon, C. Nunn, B. Stolz, C. Bruns, G. Weckbecker, I. Lewis, T. Troxler, K. Hurth, and D. Hoyer, "Drug design at peptide receptors," *Journal of Molecular Neuroscience*, vol. 18, pp. 15–27, 2002. 10.1385/JMN:18:1-2:15.

[89] D. N. Gerding, C. A. Muto, and R. C. Owens, Jr., "Measures to control and prevent clostridium difficile infection," *Clinical Infectious Diseases*, vol. 46, pp. S43–S49, 01 2008.

[90] J. Kofoed and J.-L. Reymond, "A general method for designing combinatorial peptide libraries decodable by amino acid analysis," *Journal of Combinatorial Chemistry*, vol. 9, pp. 1046–1052, 10 2007.

[91] V. S. Fluxa and J.-L. Reymond, "On-bead cyclization in a combinatorial library of 15,625 octapeptides," *Bioorganic & Medicinal Chemistry*, vol. 17, pp. 1018–1025, 2 2009.

[92] D. Berkovich-Berger and N. Gabriel Lemcoff, "Facile acetal dynamic combinatorial library," *Chemical Communications*, no. 14, pp. 1686–1688, 2008.

[93] T. Liu, S. H. Joo, J. L. Voorhees, C. L. Brooks, and D. Pei, "Synthesis and screening of a cyclic peptide library: Discovery of small-molecule ligands against human prolactin receptor," *Bioorganic & Medicinal Chemistry*, vol. 17, pp. 1026–1033, 2 2009.

[94] Y. Zhang, S. Zhou, A.-S. Wavreille, J. DeWille, and D. Pei, "Cyclic peptidyl inhibitors of Grb2 and tensin SH2 domains identified from combinatorial libraries," *Journal of Combinatorial Chemistry*, vol. 10, pp. 247–255, 02 2008.

[95] Y. Feng, A. R. Carroll, D. M. Pass, J. K. Archbold, V. M. Avery, and R. J. Quinn, "Polydiscamides bd from a marine sponge Ircinia sp. as potent human sensory neuron-specific g protein coupled receptor agonists," *Journal of Natural Products*, vol. 71, pp. 8–11, 12 2007.

[96] R. G. Linington, D. J. Edwards, C. F. Shuman, K. L. McPhail, T. Matainaho, and W. H. Gerwick, "Symplocamide A, a potent cytotoxin and chymotrypsin inhibitor from the marine Cyanobacterium Symploca sp.," *Journal of Natural Products*, vol. 71, pp. 22–27, 12 2007.

[97] K. Shimokawa, I. Mashima, A. Asai, T. Ohno, K. Yamada, M. Kita, and D. Uemura, "Biological activity, structural features, and synthetic studies of ()-ternatin, a potent fat-accumulation inhibitor of 3t3-l1 adipocytes," *Chemistry –An Asian Journal*, vol. 3, no. 2, pp. 438–446, 2008.

[98] N. Shindoh, M. Mori, Y. Terada, K. Oda, N. Amino, A. Kita, M. Taniguchi, K.-Y. Sohda, K. Nagai, Y. Sowa, Y. Masuoka, M. Orita, M. Sasamata, H. Matsushime, K. Furuichi, and T. Sakai, "YM753, a novel histone deacetylase inhibitor, exhibits antitumor activity with selective, sustained accumulation of acetylated histones in tumors in the WiDr xenograft model," *International Journal of Oncology*, vol. 32, no. 3, pp. 545–555, 2008.

[99] L. C. M. Ngoka and M. L. Gross, "Multistep tandem mass spectrometry for sequencing cyclic peptides in an ion-trap mass spectrometer," *Journal of the American Society for Mass Spectrometry*, vol. 10, pp. 732–746, 8 1999.

[100] A. Jegorov, B. Paizs, M. Zabka, M. Kuzma, V. Havlicek, A. E. Giannakopulos, and P. J. Derrick, "Profiling of cyclic hexadepsipeptides roseotoxins synthesized in vitro and in vivo: a combined tandem mass spectrometry and quantum chemical study," *European Journal of Mass Spectrometry*, vol. 9, no. 2, pp. 105–116, 2003.

[101] J. Yagüe, A. Paradela, M. Ramos, S. Ogueta, A. Marina, F. Barahona, J. López de Castro, and J. Vázquez, "Peptide rearrangement during quadrupole ion trap fragmentation: Added complexity to MS/MS spectra," *Analytical Chemistry*, vol. 75, pp. 1524–1535, 02 2003.

[102] A. Jegorov, B. Paizs, M. Kuzma, M. Zabka, Z. Landa, M. Sulc, M. P. Barrow, and V. Havlicek, "Extraribosomal cyclic tetradepsipeptides beauverolides: profiling and modeling the fragmentation pathways," *Journal of Mass Spectrometry*, vol. 39, no. 8, pp. 949–960, 2004.

[103] A. G. Harrison, A. B. Young, C. Bleiholder, S. Suhai, and B. Paizs, "Scrambling of sequence information in collision-induced dissociation of peptides," *Journal of the American Chemical Society*, vol. 128, pp. 10364–10365, 07 2006.

[104] C. Jia, W. Qi, and Z. He, "Cyclization reaction of peptide fragment ions during multistage collisionally activated decomposition: An inducement to lose internal amino-acid residues," *Journal of the American Society for Mass Spectrometry*, vol. 18, pp. 663–678, 4 2007.

[105] W. Qi, C.-X. Jia, Z.-M. He, and B. Qiao, "Multistep tandem mass spectrometry for sequencing cyclic peptides axinastatin 1 and deducing fragmentation pathways," *Huaxue Xuebao*, vol. 65, no. 3, pp. 233–238, 2007.

[106] S. Tilvi and C. G. Naik, "Tandem mass spectrometry of kahalalides: identification of two new cyclic depsipeptides, kahalalide r and s from elysia grandifolia," *Journal of Mass Spectrometry*, vol. 42, no. 1, pp. 70–80, 2007.

[107] C. Bleiholder, S. Osburn, T. D. Williams, S. Suhai, M. Van Stipdonk, A. G. Harrison, and B. Paizs, "Sequence-scrambling fragmentation pathways of protonated peptides," *Journal of the American Chemical Society*, vol. 130, pp. 17774–17789, 12 2008.

[108] A. G. Harrison, "Peptide sequence scrambling through cyclization of b5 ions," *Journal of the American Society for Mass Spectrometry*, vol. 19, pp. 1776–1780, 12 2008.

[109] I. Riba-Garcia, K. Giles, R. H. Bateman, and S. J. Gaskell, "Studies of peptide a- and b-type fragment ions using stable isotope labeling and integrated ion mobility/tandem mass spectrometry," *Journal of the American Society for Mass Spectrometry*, vol. 19, pp. 1781–1787, 12 2008.

[110] I. Riba-Garcia, K. Giles, R. H. Bateman, and S. J. Gaskell, "Evidence for structural variants of a- and b-type peptide fragment ions using combined ion mobility/mass spectrometry," *Journal of the American Society for Mass Spectrometry*, vol. 19, pp. 609–613, 4 2008.

[111] S. Jagannath and V. Sabareesh, "Peptide fragment ion analyser (PFIA): a simple and versatile tool for the interpretation of tandem mass spectrometric data and de novo sequencing of peptides," *Rapid Communications in Mass Spectrometry*, vol. 21, no. 18, pp. 3033–3038, 2007.

[112] M. Vingron, L. Wong, N. Bandeira, J. Ng, D. Meluzzi, R. Linington, P. Dorrestein, and P. Pevzner, *De Novo Sequencing of Nonribosomal Peptides*, vol. 4955, pp. 181–195. Springer Berlin / Heidelberg, 2008.

[113] S. M. Lin, L. Zhu, A. Q. Winter, M. Sasinowski, and W. A. Kibbe, "What is mzXML good for?," *Expert Review of Proteomics*, vol. 2, pp. 839–845, 11 2005.

[114] P. G. A. Pedrioli, J. K. Eng, R. Hubley, M. Vogelzang, E. W. Deutsch, B. Raught, B. Pratt, E. Nilsson, R. H. Angeletti, R. Apweiler, K. Cheung, C. E. Costello, H. Hermjakob, S. Huang, R. K. Julian, E. Kapp, M. E. McComb, S. G. Oliver, G. Omenn, N. W. Paton, R. Simpson, R. Smith, C. F. Taylor, W. Zhu, and R. Aebersold, "A common open representation of mass spectrometry data and its application to proteomics research," *Nat Biotech*, vol. 22, pp. 1459–1466, 11 2004.

[115] L. C. M. Ngoka and M. L. Gross, "A nomenclature system for labeling cyclic peptide fragments," *Journal of the American Society for Mass Spectrometry*, vol. 10, pp. 360–363, 4 1999.

[116] B. Paizs and S. Suhai, "Fragmentation pathways of protonated peptides," *Mass Spectrometry Reviews*, vol. 24, no. 4, pp. 508–548, 2005.

[117] K. Eckart, "Mass spectrometry of cyclic peptides," *Mass Spectrometry Reviews*, vol. 13, no. 1, pp. 23–55, 1994.

[118] E. Pittenauer, M. Zehl, O. Belgacem, E. Raptakis, R. Mistrik, and G. Allmaier, "Comparison of CID spectra of singly charged polypeptide antibiotic precursor ions obtained by positive-ion vacuum MALDI IT/RTOF and TOF/RTOF, AP-MALDI-IT and ESI-IT mass spectrometry," *Journal of Mass Spectrometry*, vol. 41, no. 4, pp. 421–447, 2006.

[119] F. Kopp and M. A. Marahiel, "Macrocyclization strategies in polyketide and nonribosomal peptide biosynthesis," *Natural Product Reports*, vol. 24, no. 4, pp. 735–749, 2007.

[120] M. K. Renner, Y.-C. Shen, X.-C. Cheng, P. R. Jensen, W. Frankmoelle, C. A. Kauffman, W. Fenical, E. Lobkovsky, and J. Clardy, "Cyclomarins AC, new antiinflammatory cyclic peptides produced by a marine bacterium (Streptomyces sp.)," *Journal of the American Chemical Society*, vol. 121, pp. 11273–11276, 11 1999.

[121] S. Selim, J. Negrel, C. Govaerts, S. Gianinazzi, and D. van Tuinen, "Isolation and partial characterization of antagonistic peptides produced by Paenibacillus sp. Strain B2 isolated from the Sorghum Mycorrhizosphere," *Applied and Environmental Microbiology*, vol. 71, pp. 6501–6507, 11 2005.

[122] S. S. Nair, J. Romanuka, M. Billeter, L. Skjeldal, M. R. Emmett, C. L. Nilsson, and A. G. Marshall, "Structural characterization of an unusually stable cyclic peptide, kalata B2 from Oldenlandia affinis," *Biochimica et Biophysica Acta (BBA) - Proteins & Proteomics*, vol. 1764, pp. 1568–1576, 10 2006.

[123] H. Greve, S. Kehraus, A. Krick, G. Kelter, A. Maier, H.-H. Fiebig, A. D. Wright, and G. M. König, "Cytotoxic bastadin 24 from the australian Sponge Ianthella quadrangulata," *Journal of Natural Products*, vol. 71, pp. 309–312, 02 2008.

[124] B. Adams, P. Pörzgen, E. Pittman, W. Y. Yoshida, H. E. Westenburg, and F. D. Horgen, "Isolation and structure determination of Malevamide E, a dolastatin 14 analogue, from the marine Cyanobacterium Symploca laeteviridis," *Journal of Natural Products*, vol. 71, pp. 750–754, 03 2008.

[125] J. Ng, N. Bandeira, W.-T. Liu, M. Ghassemian, T. L. Simmons, W. H. Gerwick, R. Linington, P. C. Dorrestein, and P. A. Pevzner, "Dereplication and de novo sequencing of nonribosomal peptides," *Nat Meth*, vol. 6, pp. 596–599, 08 2009.