UNIVERSITY OF CALIFORNIA, SAN DIEGO

Statistical Algorithms for High-throughput Biological Data

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

in

Electrical Engineering (Communication Theory and Systems)

by

Kyowon Jeong

Committee in charge:

Professor Pavel Pevzner, Chair
Professor Young-Han Kim, Co-Chair
Professor Vineet Bafna
Professor Nuno Bandeira
Professor Alon Orlitsky
Professor Alexander Vardy

2013

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

_____

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2013

DEDICATION

To Hanbyul and Seoyun

EPIGRAPH

子曰 學而不思則罔 思而不學則殆

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my advisor, Prof. Pavel Pevzner. Not only he provided me excellent insights and ideas for my research but also did he teach me how to communicate with other researchers efficiently and effectively. Without his brilliant guidance, this dissertation would not have been possible.

I am also thankful to Dr. Sangtae Kim, who introduced me the field of Bioinformatics and persistently helped me to adopt myself to the field. Prof. Nuno Bandeira and Prof. Vineet Bafna also have been great collaborators and they gave me constructive comments and warm encouragement which I deeply appreciate. I am grateful to Prof. Young-Han Kim, Prof. Alon Orlitsky, and Prof. Alexander Vardy for their serving in my committee.

In addition, I would like to thank Sangwoo Kim, Hosein Hosein Mohimani, and Julio Ng for their help in various projects during my graduate career. My heartfelt appreciation also goes to all members in my research group; they have been great colleagues and good friends. Finally, I want to thank all my collaborators and coauthors of my scientific publications.

Chapter 2, in full, was published as "UniNovo : a universal tool for *de novo* peptide sequencing". K. Jeong, S. Kim, and P. A. Pevzner. *Bioinformatics*, doi: 10.1093/bioinformatics/btt338. The dissertation author was the primary author of this paper.

Chapter 3 was published as "Gapped Spectral Dictionaries and Their Applications for Database Searches of Tandem Mass Spectra". K. Jeong, S. Kim, N. Bandeira, and P. A. Pevzner. *Molecular&Cellular Proteomics*, vol. 10, M110.002220, 03 2011. The dissertation author is the primary author of this paper.

Chapter 4, in full, was published as "False discovery rates in spectral identification". K. Jeong, S. Kim, and N. Bandeira. *BMC Bioinformatics*, vol. 13(Suppl 16), S2, 11 2012. The dissertation author is the primary author of this paper.

Chapter 5, in full, was submitted as "Virmid: virtual microdissection of sample mixtures for accurate somatic mutation profiling". S. Kim, K. Jeong, K. Bhutani1, J. Lee, A. Patel, E. Scott, H. Nam, H. Lee, J. G. Gleeson, and V. Bafna. Sangwoo Kim and the dissertation author were the primary authors of this paper.

VITA

2007	B. S. in Electrical Engineering, Seoul National University, Seoul, Korea
2013	Ph. D. in Electrical Engineering (Communication Theory and Systems), University of California, San Diego

PUBLICATIONS

Kyowon Jeong, Sangtae Kim, and Pavel Pevzner, UniNovo : a universal tool for *de novo* peptide sequencing, Bioinformatics, doi: 10.1093/bioinformatics/btt338.

Sangwoo Kim, Kyowon Jeong, Kunal Bhutani, Jeong Ho Lee, Anand Patel, Eric Scott, Hojung Nam, Hayan Lee, Joseph G. Gleeson, and Vineet Bafna, Virmid: virtual microdissection of sample mixtures for accurate somatic mutation profiling, submitted.

Kyowon Jeong, Sangtae Kim, and Pavel Pevzner, UniNovo : a universal tool for *de novo* peptide sequencing, In Proceedings of the Seventeenth International Conference on Research in Computational Molecular Biology (RECOMB-2013), 100-117, 2013.

Sangwoo Kim, Kyowon Jeong, and Vineet Bafna, Wessim: a whole-exome sequencing simulator based on in silico exome capture, Bioinformatics, 29(8), 1076-1077, 2013.

Kyowon Jeong, Sangtae Kim, and Nuno Bandeira, False Discovery Rates in Spectral Identification, BMC Bioinformatics, 13(Suppl 16), S2, 2012.

Kyowon Jeong and Jungwoo Lee, Channel Parameter Tracking for Adaptive MMSE Channel Estimation in OFDM Systems, IEICE Transactions 95-A(8), 1439-1443, 2012.

Chanhong Kim, Kyowon Jeong, Kyungjun Ko, and Jungwoo Lee, SNR-based adaptive modulation for wireless LAN systems, In Proceedings of of the IEEE International Symposium on Circuits and Systems (ISCAS), 758-761, 2012.

Kyowon Jeong, Sangtae Kim, Nuno Bandeira, and Pavel Pevzner, Gapped Spectral Dictionaries and Their Applications for Database Searches of Tandem Mass Spectra, Molecular & Cellular Proteomics, 10, M110.002220, 2011.

Kyowon Jeong, Sangtae Kim, Nuno Bandeira, and Pavel Pevzner, Gapped Spectral Dictionaries and Their Applications for Database Searches of Tandem Mass Spectra, In Proceedings of the Fourteenth International Conference on Research in Computational Molecular Biology (RECOMB-2010), 208-232, 2010.

Kyowon Jeong and Jungwoo Lee, Low Complexity Channel Tracking for Adaptive MMSE Channel Estimation in OFDM, In Proceedings of of the Conference on Information Sciences and Systems (CISS), 548-552, 2007.

ABSTRACT OF THE DISSERTATION

Statistical Algorithms for High-throughput Biological Data

by

Kyowon Jeong

Doctor of Philosophy in Electrical Engineering (Communication Theory and Systems)

University of California, San Diego, 2013

Professor Pavel Pevzner, Chair
Professor Young-Han Kim, Co-Chair

Recent advances in high-throughput technologies, such as tandem mass spectrometry (MS/MS) and next generation sequencing (NGS), have enabled the acquisition of huge amount of biological data containing whole genome/proteome scale information. However, due to their huge size and complexity, the interpretation of such data has become the bottleneck for further biological applications; many related computational algorithms and standardized statistical methods are still missing. Therefore, the development of efficient statistical algorithms has become essential to analyze and access massive biological data. In this dissertation, statistical algorithms for the peptide identification via MS/MS spectra and the somatic mutation profiling via NGS read data are presented.

Peptide/protein identification via mass spectrometry is an important task for proteomics studies. Two most widely used approaches are the database search and the *de novo* peptide sequencing. We first present UniNovo, a universal *de novo* peptide sequencing algorithm that works well for various types of spectra from different experimental protocols and MS instrument configurations. Next we introduce MS-GappedDictionary, an algorithm that enables fast and sensitive searches of huge proteome databases (which have been prohibitively time consuming with existing approaches) using *de novo* sequences generated from tandem mass spectra. Lastly we present a statistical method to validate the accuracy of false discovery rate (FDR) estimation in database searches and suggest a standard method for more accurate estimation of FDRs.

The later part of this dissertation focuses on the somatic mutation profiling via NGS read data. The goal of the somatic mutation profiling is to identify genetic alterations that occur after conception, or somatic mutations. Since the somatic mutations can (but not always) cause cancer or other diseases, their identification is crucial for downstream disease studies. However, sensitive identification of somatic mutations is a hard task because they are extremely rare events (1-10 occurrences per 1 Mega base pairs). We introduce a novel algorithm for identifying somatic mutations which incorporates the possible contamination of biological samples into the model. Using both simulated and experimental datasets, we demonstrate that our algorithm has higher sensitivity than other state-of-the-art algorithms.

Chapter 1

Introduction

Recent advances in high-throughput technologies, such as tandem mass spectrometry (MS/MS) and next generation sequencing (NGS), have enabled the acquisition of huge amount of biological data containing whole genome/proteome scale information. However, due to their huge size and complexity, the interpretation of such data has become the bottleneck for further biological applications; many related computational algorithms and standardized statistical methods are still missing. Therefore, the development of efficient statistical algorithms has become essential to analyze and access massive biological data. In this dissertation, statistical algorithms for the peptide identification via MS/MS spectra (Chapter 2, 3, and 4) and the somatic mutation profiling via NGS read data (Chapter 5) are presented.

1.1 Universal *de novo* peptide sequencing algorithm

Mass spectrometry (MS) instruments and experimental protocols are rapidly advancing, but *de novo* peptide sequencing algorithms to analyze tandem mass (MS/MS) spectra are lagging behind. While existing *de novo* sequencing tools perform well on certain types of spectra (e.g., Collision Induced Dissociation (CID) spectra of tryptic peptides), their performance often deteriorates on other types of spectra, such as Electron Transfer Dissociation (ETD), Higher-energy Collisional Dissociation (HCD) spectra, or spectra of non-tryptic digests. In Chapter 2, we present a *universal*

de novo sequencing algorithm called UniNovo that works well for all types of spectra or even for spectral pairs (e.g., CID/ETD spectral pairs). UniNovo uses an improved scoring function that captures the dependences between different ion types, where such dependencies are learned automatically using a modified offset frequency function [DAC+99]. The performance of UniNovo is compared with PepNovo+, PEAKS, and pNovo using various types of spectra. The results show that the performance of UniNovo is superior to other tools for ETD spectra and superior or comparable to others for CID and HCD spectra. UniNovo also estimates the probability that each reported reconstruction is correct, using simple statistics that are readily obtained from a small training dataset. We demonstrate that the estimation is accurate for all tested types of spectra (including CID, HCD, ETD, CID/ETD, and HCD/ETD spectra of trypsin, LysC, or AspN digested peptides).

1.2 Enabling proteogenomic searches in six-frame translation of genomic sequences

Generating all plausible *de novo* interpretations of a peptide tandem mass (MS/MS) spectrum (Spectral Dictionary) and quickly matching them against the database represent a recently emerged alternative approach to peptide identification. However, the sizes of the Spectral Dictionaries quickly grow with the peptide length making their generation impractical for long peptides. In Chapter 3, we introduce Gapped Spectral Dictionaries (all plausible *de novo* interpretations with gaps) that can be easily generated for any peptide length thus addressing the limitation of the Spectral Dictionary approach. We show that Gapped Spectral Dictionaries are small thus opening a possibility of using them to speed-up MS/MS searches. Our MS-GappedDictionary algorithm (based on Gapped Spectral Dictionaries) enables proteogenomics applications (like searches in the six-frame translation of the human genome) that are prohibitively time consuming with existing approaches. MS-GappedDictionary generates gapped peptides that occupy a niche between accurate but short peptide sequence tags and long but inaccurate full length peptide reconstructions. We show that, contrary to conventional wisdom, some high quality

spectra do not have good peptide sequence tags and introduce gapped tags that have advantages over the conventional peptide sequence tags in MS/MS database searches.

1.3 False discovery rates in spectral identification

Automated database search engines are one of the fundamental engines of high-throughput proteomics enabling daily identifications of hundreds of thousands of peptides and proteins from tandem mass (MS/MS) spectrometry data. Nevertheless, this automation also makes it humanly impossible to manually validate the vast lists of resulting identifications from such high-throughput searches. This challenge is usually addressed by using a Target-Decoy Approach (TDA) to impose an empirical False Discovery Rate (FDR) at a pre-determined threshold $x\%$ with the expectation that at most $x\%$ of the returned identifications would be false positives. But despite the fundamental importance of FDR estimates in ensuring the utility of large lists of identifications, there is surprisingly little consensus on exactly how TDA should be applied to minimize the chances of biased FDR estimates. In fact, since less rigorous TDA/FDR estimates tend to result in more identifications (at higher 'true' FDR), there is often little incentive to enforce strict TDA/FDR procedures in studies where the major metric of success is the size of the list of identifications and there are no follow up studies imposing hard cost constraints on the number of reported false positives.

In Chapter 4, we address the problem of the accuracy of TDA estimates of empirical FDR. Using MS/MS spectra from samples where we were able to define a *factual* FDR estimator of 'true' FDR we evaluate several popular variants of the TDA procedure in a variety of database search contexts. We show that the fraction of false identifications can sometimes be over $10\times$ higher than reported and may be unavoidably high for certain types of searches. In addition, we further report that the two-pass search strategy seems the most promising database search strategy.

While unavoidably constrained by the particulars of any specific evaluation dataset, our observations support a series of recommendations towards maximizing the number of resulting identifications while controlling database searches with robust and reproducible TDA estimation of empirical FDR.

1.4 Sensitive somatic mutation profiling incorporating sample impurity

Genetic alterations that occur after conception, or somatic mutations, are rare events as 1-10 occurrences per 1 Mbp, and can be only detected when matched control is provided with disease sample. While advanced sample preparation technologies have enabled accurate acquisition of designated cell types, there are many cases where contamination of disease samples from surrounding control cells is unavoidable. With this sample impurity, variant calling models are severely compromised fail to identify many true mutation events.

In Chapter 5, we present a novel probabilistic model, Virmid, to estimate the proportion of control (i.e. healthy cells) in the mixed disease sample (α) and to use it for more accurate somatic mutation profiling. Given a paired disease-control dataset, Virmid estimates α and joint genotype probabilities using maximum likelihood estimator (MLE) while minimizing observational biases. In a mixed sample set simulated from human chromosome 1, Virmid could estimate the true value α to within 0.5% error over a wide range of values, including extreme distributions ($\alpha \sim 1\%, \alpha \sim 90\%$). Parameterizing over estimated α dramatically improved the overall accuracy of somatic mutation prediction (up to 10-fold increase in area under precision-recall curves). In a test on 15 public breast cancer data, Virmid showed the best sensitivity (98.6%) within a very satisfactory specificity (< 1 false call in a million base-pair). When applied to a recently sequenced hemimegalencephaly exome data (n=5), Virmid estimated α as 64%-85% and identified more than 2000 novel somatic mutations including 923 missense events.

We showed that Virmid accurately estimates sample contamination and improves mutation finding. Virmid can be utilized in many cases where sample purity is questionable.

Chapter 2

UniNovo : a universal tool for *de novo* peptide sequencing

2.1 Introduction

De novo peptide sequencing by tandem mass (MS/MS) spectrometry is a valuable alternative to MS/MS database search. In contrast to the database search approach that utilizes the information from proteome, the *de novo* sequencing approach attempts to identify peptides only using the information from the input spectrum. Hence, most *de novo* sequencing algorithms are based on the prior knowledge of the fragmentation characteristics (e.g., ion types and their propensities) of MS/MS spectra [MZH$^+$03, FP05, Fra09].

The fragmentation characteristics are highly dependent on the fragmentation method used to generate the spectrum. Among several fragmentation methods available, the collision induced dissociation (CID) is the most commonly used method. Accordingly, the fragmentation characteristics of CID have been well studied compared to recently introduced fragmentation methods, such as electron transfer dissociation (ETD) and higher-energy collisional dissociation (HCD) [JMB$^+$87, WTSB00, BTYW03, THWY04, HTT$^+$05, BW09]. As a result, many *de novo* sequencing algorithms have been introduced for CID spectra; for example, PEAKS [MZH$^+$03] and PepNovo+ [FP05, Fra09], are the state of the art *de novo* sequencing tools for CID spectra.

Other fragmentation methods like ETD and HCD have a great potential for *de novo* sequencing. For example, for highly charged spectra, ETD provides better fragmentation and thus is better suited for *de novo* sequencing than CID [ZZS08, SMC08]. Also, more complete fragmentation of peptide ions (especially in low mass regions) in HCD provides a better chance to obtain more accurate *de novo* reconstructions than CID [OML$^+$07, CSY$^+$10]. Furthermore, modern mass spectrometers (e.g., LTQ-Orbitrap Velos) allow the generation of paired spectra (e.g., CID/ETD or HCD/ETD spectral pairs). Since CID (or HCD) and ETD spectra provide complementary information for peptide sequencing [SNKZ05, DB09, HM10], such spectral pairs (or even triplets) enable more accurate *de novo* sequencing.

Several *de novo* sequencing algorithms were recently presented to take advantage of those new fragmentation methods. For instance, [LSXM10] proposed a *de novo* sequencing algorithm for ETD spectra, which is used by PEAKS. For HCD spectra, [CSY$^+$10] introduced a *de novo* sequencing tool, pNovo, that not only takes advantage of the high precision peaks in HCD spectra but also uses the information of abundant immonium and internal ions. In case of spectral pairs, [SNKZ05] proposed a greedy algorithm (for CID/ECD spectral pairs) that significantly boosts the performance of *de novo* sequencing. [DB09] presented Spectrum Fusion, a *de novo* sequencing algorithm for CID/ETD spectral pairs. Spectrum Fusion constructs a combined spectrum from the input CID/ETD spectral pair using a Bayesian Network. It generates multiple *de novo* sequences using the combined spectrum and score them by the scoring function in ByOnic [BCG07]. [HM10] also presented a *de novo* sequencing algorithm, ADEPTS, for CID/ETD spectral pairs. Given a CID/ETD spectral pair, ADEPTS first finds 1,000 candidate *de novo* sequences from each spectrum, using PEAKS. The total 2,000 candidate sequences are then rescored against the input spectral pair, and the best-scoring peptide is reported.

While the above tools perform well for the spectra generated from the fragmentation method(s) that each tool targeted, they often generate inferior results for the spectra from other fragmentation methods. Moreover, if alternative proteases (e.g., LysC or AspN) are used for protein digestion, these tools may produce suboptimal results because different proteases often generate peptides with different fragmentation characteristics [KMB$^+$10].

In case of the database search approach, [KMB+10] recently introduced a universal algorithm MS-GFDB that shows a significantly better peptide identification performance than other existing database search tools such as Mascot +Percolator [PPCC99, KCW+07]. However, a universal *de novo* sequencing tool is still missing.

We present UniNovo, a universal *de novo* sequencing tool that can be generalized for various *types* (i.e., the combinations of the fragmentation method and the protease used to digest sample proteins) of spectra. The scoring function of UniNovo is easily trainable using a training dataset consisting of thousands of annotated spectra. All information needed for *de novo* sequencing are learned from the training dataset, and the running time for training is less than 5 hours in a typical desktop environment. Currently UniNovo is trained for CID, HCD, and ETD spectra of trypsin, LysC, or AspN digested peptides. We show that the performance of UniNovo is better than or comparable to PepNovo+, PEAKS, and pNovo for various types of spectra.

One of the biggest challenges in *de novo* sequencing is to estimate the error rate of the resulting *de novo* reconstructions. Unlike MS/MS database search tools that commonly uses the *target-decoy approach* [EG07, Nes10] to estimate the statistical significance of the peptide identifications, *de novo* reconstructions have rarely been subjected to a statistical significance analysis in the past.

Several *de novo* sequencing tools report the error rate of amino acid predictions (e.g. confidence scores in PEAKS), but this is often not sufficient because the overall quality of the sequence cannot be easily determined by the error rates of individual amino acid predictions. To our knowledge, only PepNovo+ reports the empirical probability that the output peptide is correct. PepNovo+ predicts the probabilities using logistic regression with multiple features of the reconstructions such as length and score, which are extracted from a training dataset consisting of hundreds of thousands of annotated spectra [Fra09]. However, PepNovo+ does not include an automated training procedure (that would allow to easily extend PepNovo+ for newly emerging mass spectrometry approaches) and is currently trained only for CID.[1] Thus, in case of non-CID fragmentation methods, it remains unclear how to obtain accurate

[1]Extending PepNovo+ beyond CID spectra requires training complex boosting-based re-ranking models for predicting peak ranks and rescoring peptide candidates. PepNovo+ training includes several manual steps and the availability of a very large corpus of training spectra (Ari Frank, personal communication, October 5, 2012).

error rate estimation for *de novo* reconstructions.

UniNovo estimates the probability that each reported reconstruction is correct, using simple statistics that are readily obtained from a small training dataset. We demonstrate that the estimation is accurate for all tested types of spectra (including CID, HCD, ETD, CID/ETD, and HCD/ETD spectra of trypsin, LysC, or AspN digested peptides). This allows UniNovo to automatically filter out low quality spectra.

2.2 Methods

Similar to [KGBP09], we first describe the algorithm on a simplified model that assumes the following:

- the masses of amino acids are integers (e.g., the mass of Gly is 57).
- the m/z (mass to charge ratio) of peaks (in spectra) are integers.
- the intensity of all peaks is 1.
- only N-terminal charge 1 ions are considered (e.g., b, c, or $b - H_2O$ ions, but not y-ion series).
- the *parent mass* (the mass of the precursor ion) of a spectrum equals to the mass of the peptide that generated the spectrum.

While such a simplified model is impractical, we chose to introduce our algorithm on this model for better understanding of the algorithm on a more complex and realistic model. The algorithm on a more realistic model is followed by the algorithm on the simple model.

2.2.1 Vector operations

We first define the following vector operations. Let V and W be Boolean vectors with n elements.

- $|V|$ is the number of elements in V (i.e., n).
  $\Rightarrow$ For $V = (0, 1, 0, 1, 0)$, $|V| = 5$.

- $\langle V \rangle$ is the number of non-zero elements in V.
  $\Rightarrow$ For $V = (0, 1, 0, 1, 0)$, $\langle V \rangle = 2$.

- $V \cdot W$ denotes the elementwise multiplication between V and W.
  $\Rightarrow$ For $V = (0, 1, 0, 1, 0)$ and $W = (1, 1, 1, 0, 0)$, $V \cdot W = (0, 1, 0, 0, 0)$.

- Given an integer k, a vector V^k is a vector obtained by shifting all elements of V by k. More formally, V^k is a vector of cardinality n whose elements are given by

$$V^k(i) = \begin{cases} V(i-k) & \text{if } 1 \leq i-k \leq n \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

  for $i = 1, \cdots, n$.
  $\Rightarrow$ For $V = (0, 1, 0, 1, 0)$, $V^{-2} = (0, 1, 0, 0, 0)$ and $V^{+1} = (0, 0, 1, 0, 1)$.

2.2.2 Terminology and definitions

Let A be the set of amino acids with (integer) masses $m(a)$ for $a \in A$. A *peptide* $a_1 a_2 \cdots a_k$ is a sequence of amino acids, and the mass of a peptide is the total mass of amino acids in the peptide. We represent a peptide $a_1 a_2 \cdots a_k$ with mass n by a Boolean vector $P = (P_1, \cdots, P_n)$, where $P_i = 1$ if $i = \sum_{t=1}^{j} a_t$ for $0 < j < k$, and $P_i = 0$ otherwise. If $P_i = 1$, we call a mass i a *fragmentation site*. For example, suppose there are two amino acids A and B with masses 2 and 3, respectively. Then, the peptide $ABBA$ has the mass of $2+3+3+2 = 10$ and is represented by a Boolean vector $(0, 1, 0, 0, 1, 0, 0, 1, 0, 0)$. The fragmentation sites of this peptide are, thus, 2, 5, and 8.

A *spectrum* is a list of peaks, where each peak is specified by an m/z. We represent a spectrum of parent mass n by a Boolean vector $S = (S_1, \cdots, S_n)$, where $S_i = 1$ if the peak of m/z i (or simply the peak i) is present and $S_i = 0$ otherwise.[2]

A *peptide-spectrum match (PSM)* is a pair (P, S) formed by a peptide P and a spectrum S. Given an integer δ called an *ion type* and a PSM (P, S), we say a peak i is a *δ-ion peak* (with respect to P) if $i - \delta$ is a fragmentation site, that is, $P_{i-\delta} = 1$.

[2]Representing peptides and spectra as vectors allows us to represent the generation of spectra from peptides by simple vector operations.

In this model, the ion type can be any integer. In the connection to the experimental MS/MS spectra, ion types can represent common singly charged N-terminal ions; for example, the ion types 1 and -27 represent b and a ions, respectively.

Given an integer f called a *feature* and a spectrum S, we say that a peak i *satisfies* f if another peak $i + f$ is present in the spectrum, that is, $S_{i+f} = 1$. For instance, a peak 30 satisfies a feature $f = -18$, if $S_{30-18} = 1$. In experimental spectra, various ions are often observed along with neutral losses (e.g., b-ion and $b - H_2O$-ion) or with related ions (e.g., b-ion and a-ion). A feature describes the relation (the shift of m/z values in this simplified model) between two peaks that may correspond to a neutral loss or a mass gain/loss between related ions. For example, since we are dealing with only charge 1 ions, a water loss (from any ions) is represented by the feature $f = -18$, and the mass gain from a-ion to b-ion is represented by the feature $f = +27$.

2.2.3 Peptide-spectrum generative model

We model how a peptide P of mass n generates a spectrum S. Apart from a 1-step generative model in [BOMP08] or [KGBP09], we introduce a more adequate 2-step probabilistic model in which the dependency between different ions can be described.

Assume that we are given the set of ion types (the *ion type set* Δ) and the set of features (the *feature set* F). For simplicity, we consider the case where only one ion type $\delta = 0$ is in Δ and one feature f is in F. Given a peptide P, a *partial-spectrum* s is generated per each element P_i of P as follows: The probability that $s_i = 1$ is given by α if $P_i = 1$ or by β otherwise (the first generation step). This first step can be characterized by a 2×2 matrix called the *ion type matrix* (Figure 2.1). When $s_i = 1$, the probability that $s_{i+f} = 1$ (i.e., the peak i satisfies f) is given by μ if $P_i = 1$ or ν otherwise (the second generation step). The second step is characterized by the *feature-ion type matrix* (Figure 2.1).[3] The second step can describe the dependency between different ions (or an ion and its neutral loss) from the same fragmentation site. If multiple ion types and multiple features are considered, the ion type matrix

[3]Given $s_i = 0$, the probability that $s_{i+f} = 1$ is assumed to be 0.

should be defined per ion type, and the feature-ion type matrix per ion type and per feature. The spectrum S is generated by taking elementwise OR operation for the generated partial-spectra s.

2.2.4 Training UniNovo

Since the ion type matrices and feature-ion type matricies fully describe the generation of a spectrum, in the training step, UniNovo learns these matrices from the *training dataset* $\mathcal{T}$ (a set of PSMs). The training of UniNovo consists of two stages: ion type selection and feature detection.

Ion type selection

In the ion type selection step, the frequently observed ion types are selected from the training dataset using the *offset frequency function* (OFF) introduced by Dancik et al. 1999 [DAC+99]. And the probabilities in the ion type matrices for the ion types (e.g., α and β in Figure 2.1 (a)) are learned.

Given an ion type δ, OFF outputs the empirical probability that a δ-ion peak is observed for a fragmentation site in the training dataset. We define OFF as follows: The input to OFF is the training dataset $\mathcal{T}$. OFF is defined by

$$OFF(\delta) = \frac{\sum_{(P,S)\in\mathcal{T}} \overbrace{\langle S \cdot P^{\delta} \rangle}^{\#\ \delta\text{-ion peaks in } S}}{\sum_{(P,S)\in\mathcal{T}} \underbrace{\langle P \rangle}_{\#\ \text{fragmentation sites in } P}}. \tag{2.2}$$

Out of all ion types δ satisfying $-38 < \delta < 38$, we pick 8 ion types δ with the highest values of $OFF(\delta)$. We denote the set of the selected ion types as *ion type set* Δ.

After learning the ion type set Δ, we learn α and β for each ion type in Δ. Given an ion type δ, α is simply given by $\alpha := OFF(\delta)$. β can be obtained by

$$\beta = \frac{\sum_{(P,S)\in\mathcal{T}} \overbrace{\langle S \rangle - \langle S \cdot P^{\delta} \rangle}^{\#\ \text{non-}\delta\text{-ion peaks in } S}}{\sum_{(P,S)\in\mathcal{T}} \underbrace{|P| - \langle P \rangle}_{\#\ \text{non-fragmentation sites in } P}}. \tag{2.3}$$

We also learn the empirical probability that a random mass i is a fragmentation site, i.e., $Pr(P_i = 1)$. To learn this probability, first an element in the peptide of each PSM is selected randomly. The probability is estimated by the frequency of the selected elements being fragmentation sites. The learned probability is called *prior fragmentation probability* and is denoted by p.[4]

Feature detection step

The feature detection step aims to detect the features that the peaks of the ion types in Δ often satisfy. Besides, the probabilities in the feature-ion type matrices (μ and ν in Figure 2.1 (a)) are learned.

The features are detected using OFF with a slight modification, which is called a *feature frequency function (FFF)*. Given an ion type δ and a feature f, FFF outputs the empirical probability that a δ-ion peak satisfies f.

The inputs to FFF are the training dataset $\mathcal{T}$, an ion type δ, and a feature f. FFF for $\delta \in \Delta$ is defined by

$$FFF(\delta, f) = \frac{\sum_{(P,S)\in\mathcal{T}} \overbrace{\langle S \cdot S^{-f} \cdot P^{\delta} \rangle}^{\#\ \delta\text{-ion peaks satisfying } f \text{ in } S}}{\sum_{(P,S)\in\mathcal{T}} \underbrace{\langle S \cdot P^{\delta} \rangle}_{\#\ \delta\text{-ion peaks in } S}}. \tag{2.4}$$

We select all features f such that $FFF(\delta, f) > 0.15$ and $-38 < x < 38$ for $\delta \in \Delta$. The selected features are called an *offset features*. Since the features satisfying $FFF(\delta, f) > 0.15$ are selected regardless of the size of the feature set, the total number of features in UniNovo is not fixed. In general, the total number was about several thousands.

In addition, the features $f = m(a), a \in A$ are selected, and the selected features are called *linking features*. A linking feature characterizes two peaks whose mz difference equals to an amino acid mass. The set of selected offset and linking features is named as the *feature set* and is denoted by F.

[4]When masses of amino acids are rounded to integers, $Pr(P_i = 1) \approx \frac{1}{121.6}$. However, if we consider more accurate amino acid masses (for the spectra of high resolution), this probability should be learned from the training dataset.

Given an ion type $\delta \in \Delta$ and a feature $f \in F$, we learn μ and ν. μ is simply given by $FFF(\delta, f)$ whereas ν is given by

$$\nu = \frac{\sum_{(P,S)\in\mathcal{T}} \overbrace{\langle S \cdot S^{-f} \rangle - \langle S \cdot S^{-f} \cdot P^{\delta} \rangle}^{\#\text{ non-}\delta\text{-ion peaks satisfying } f \text{ in } S}}{\sum_{(P,S)\in\mathcal{T}} \underbrace{\langle S \rangle - \langle S \cdot P^{\delta} \rangle}_{\#\text{ non-}\delta\text{-ion peaks in } S}}. \tag{2.5}$$

We emphasize that all the above probabilities can be learned from a small set of PSMs even if there are many ion types in Δ and features in F because each probability is associated to an individual ion type or a combination of an ion type and a feature, not a combination of multiple ion types and multiple features.

2.2.5 How to infer fragmentation sites from a spectrum

Given a spectrum S of parent mass n, our goal is to predict the fragmentation sites of the (unknown) peptide P that generated S. For simplicity, assume that there exists a single ion type $\delta = 0$ is in the ion type set Δ (but multiple features in the feature set F). Given a peak i, define H as the set of features that the peak i satisfies. Then the fragmentation sites are predicted by solving the following Bayesian inference problem.

Fragmentation inference problem: Given the set of features H and P_i such that $Pr(P_i = 1) = p$ (the prior fragmentation probability), derive the posterior probability $Pr(P_i = 1 | S_i = S_{i+f} = 1 \text{ for } f \in H)$.

Since we assumed that there is only one ion type, we have only one ion type matrix. On the other hand, per each feature we have a feature-ion type matrix. Let μ_f and ν_f denote μ and ν associated to the feature f, respectively. Further assume that all features are independent (i.e., the events "$S_i = S_{i+f} = 1$ for f" are independent for $f \in H$) Then, if H is an empty set, the posterior probability $Pr(P_i = 1 | S_i = S_{i+f} = 1 \text{ for } f \in H)$ reduces to $Pr(P_i = 1 | s_i = 1)$. By Bayes's rule, we have

$$Pr(P_i = 1 | s_i = 1) \propto Pr(P_i = 1) \cdot Pr(s_i = 1 | P_i = 1) = p \cdot \alpha. \tag{2.6}$$

Similarly, we obtain $Pr(P_i = 0|s_i = 1) \propto (1-p)\cdot\beta$. Since $Pr(P_i = 1|s_i = 1)+Pr(P_i = 0|S_i^i = 1) = 1$, we obtain

$$Pr(P_i = 1|s_i = 1) = \frac{Pr(P_i = 1|s_i = 1)}{Pr(P_i = 1|s_i = 1) + Pr(P_i = 0|s_i = 1)} = \frac{p \cdot \alpha}{p \cdot \alpha + (1-p) \cdot \beta}. \tag{2.7}$$

Denote this probability as γ. Then, we obtain

$$\begin{aligned}
&Pr(P_i = 1|s_i = 1 \text{ and } s_{i+f} = 1 \text{ for } f \in H) && (2.8)\\
&\propto Pr(P_i = 1|s_i = 1) \cdot Pr(s_{i+f} = 1 \text{ for } f \in H|P_i = 1, S_i = 1) && (2.9)\\
&= \gamma \cdot Pr(s_{i+f} = 1 \text{ for } f \in H|P_i = 1, s_i = 1) && (2.10)\\
&= \gamma \cdot \prod_{f\in H} Pr(s_{i+f} = 1|P_i = 1, s_i = 1) && (2.11)\\
&= \gamma \cdot \prod_{f\in H} \mu_f && (2.12)
\end{aligned}$$

where μ_f denotes the probability μ associated to the feature f. The equality between (2.10) and (2.11) is obtained from the assumed independence between features. Likewise, we can show $Pr(P_i = 0|s_i = 1 \text{ and } s_{i+f} = 1 \text{ for } f \in H) \propto (1-\gamma) \cdot \prod_{f\in H} \nu_f$ where ν_f is the probability ν associated to the feature f. Therefore, we obtain

$$\begin{aligned}
&Pr(P_i = 1|S_i = 1 \text{ and } S_{i+f} = 1 \text{ for } f \in H) && (2.13)\\
&= Pr(P_i = 1|s_i = 1 \text{ and } s_{i+f} = 1 \text{ for } f \in H) && (2.14)\\
&= \frac{\gamma \cdot \prod_{f\in H} \mu_f}{\gamma \cdot \prod_{f\in H} \mu_f + (1-\gamma) \cdot \prod_{f\in H} \nu_f}. && (2.15)
\end{aligned}$$

Denote the obtained probability in (2.15) as π_i. We define a *fragmentation probability vector ($\mathcal{FPV}$)* as a vector with n elements such that

$$\mathcal{FPV}_i = \begin{cases} \pi_i & \text{if } S_i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

for $i = 1, \cdots, n-1$, and $\mathcal{FPV}_n := 1$ (see Figure 2.1 (b)). $\mathcal{FPV}_i$ is an estimated probability that $P_i = 1$. We use $\mathcal{FPV}$ for the generation of *de novo* reconstructions.

The equation (2.15) is based on a simplified model in which a single one ion type and multiple independent features are used. However, some features are known

to be strongly dependent each other (e.g., a feature describing a single water loss and a double water losses), and usually multiple ion types are present in the ion type set. Below we describe how to calculate $\mathcal{FPV}$ for such cases.

Multiple ion types and multiple but independent features

We consider the case in which multiple ion types are present in the ion type set Δ. For an ion type $\delta \in \Delta$, the expression (2.15) can be generalized as

$$Pr(P_{i-\delta} = 1 | S_i = 1 \text{ and } S_{i+f} = 1 \text{ for } f \in H) = \frac{\gamma_\delta \cdot \prod_{f \in H} \mu_f^\delta}{\gamma_\delta \cdot \prod_{f \in H} \mu_f^\delta + (1 - \gamma_\delta) \cdot \prod_{f \in H} \nu_f^\delta} \tag{2.17}$$

where γ_δ denotes γ of the ion type matrix for the ion type δ, and μ_f^δ (ν_f^δ) denotes μ (ν) for the feature f and the ion type δ. Denote the obtained probability in (2.17) as π_i^δ. For each ion type δ, we derive a fragmentation probability vector $\mathcal{FPV}^\delta$ as

$$\mathcal{FPV}_i^\delta = \begin{cases} \pi_{i+\delta}^\delta & \text{if } S_{i+\delta} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

for $i = 1, \cdots, n-1$. $\mathcal{FPV}_n^\delta$ is again defined to be 1. The final fragmentation probability vector $\mathcal{FPV}$ is generated by taking elementwise (weighted) summation of $\mathcal{FPV}^\delta$'s for $\delta \in \Delta$. The weights are decided by an MMSE (minimum mean squared error) estimation method as described below.

For simplicity, we start with the case in which the ion type set is given by $\Delta = \{\delta, \delta'\}$. Given a spectrum , UniNovo generates 2 fragmentation probability vectors ($\mathcal{FPV}^\delta$ and $\mathcal{FPV}^{\delta'}$), and the final $\mathcal{FPV}$ is generated by elementwise weighted summation of these fragmentation probability vectors.

The weights are learned from the training dataset $\mathcal{T}$ as follows: For each PSM (P, S) in $\mathcal{T}$, we first generate $\mathcal{FPV}^\delta$ and $\mathcal{FPV}^{\delta'}$. Given an index i, we consider three different cases for $(\mathcal{FPV}_i^\delta, \mathcal{FPV}_i^{\delta'})$: only $\mathcal{FPV}_i^\delta$ is non-zero, only $\mathcal{FPV}_i^{\delta'}$ is non-zero, and both are non-zero. The weights are learned separately for each case (and are multiplied separately for each case when we generate the final $\mathcal{FPV}$). We describe the last case (both are non-zero) only. Let $\overline{X}$ denotes the sample mean of X. For instance, $\overline{\mathcal{FPV}_i^\delta}$ denotes the sample mean of $\mathcal{FPV}_i^\delta$.

The autocorrelation matrix $\mathbf{R}$ is defined as

$$\mathbf{R} = \begin{bmatrix} \overline{\mathcal{FPV}_i^{\delta}\mathcal{FPV}_i^{\delta}} & \overline{\mathcal{FPV}_i^{\delta}\mathcal{FPV}_i^{\delta'}} \\ \overline{\mathcal{FPV}_i^{\delta'}\mathcal{FPV}_i^{\delta}} & \overline{\mathcal{FPV}_i^{\delta'}\mathcal{FPV}_i^{\delta'}} \end{bmatrix}$$

and the crosscorrelation matrix $\mathbf{C}$ is defined as

$$\mathbf{C} = \begin{bmatrix} \overline{\mathcal{FPV}_i^{\delta}P_i} \\ \overline{\mathcal{FPV}_i^{\delta'}P_i} \end{bmatrix}.$$

The weight vector is given by

$$W = \mathbf{R}^{-1}\mathbf{C}.$$

When more than two ion types are present in the ion type set $\Delta = \{\delta_1, \cdots, \delta_l\}$, UniNovo generates l fragmentation probability vectors, $\mathcal{FPV}^{\delta_1}, \cdots, \mathcal{FPV}^{\delta_l}$. The weight vectors are learned as above separately for $2^l - 1$ different cases for $(\mathcal{FPV}^{\delta_1}, \cdots, \mathcal{FPV}^{\delta_l})$: only $\mathcal{FPV}_i^{\delta_1}$ is non-zero, only $\mathcal{FPV}_i^{\delta_2}$ is non-zero, $\cdots$, all $\mathcal{FPV}_i^{\delta_1}$ to $\mathcal{FPV}_i^{\delta_l}$ are non-zero.

Multiple ion types and multiple dependent features:

The above derivations of the posterior probability are valid only if the features in H are mutually independent. However, in practice, some features are often strongly correlated (e.g., a feature describing the loss of a water molecule and another describing the loss of two water molecules). Thus, out of all features that a peak i satisfies, a few "statistically meaningful" features that are weakly correlated are automatically selected for H.

To select statistically meaningful and weakly correlated features out of H, we first define the *divergence* of a feature f. We again assume that only one ion type $\delta = 0$ is present in the ion type set Δ. If two probabilities $Pr(P_i = 1|S_i = 1)$ and $Pr(P_i = 1|S_i = S_{i+f} = 1)$ are similar to each other, we can conclude that the feature f is not helpful to determine the fragmentation sites. The two probabilities are given by $p{\cdot}\alpha$ and $\gamma{\cdot}\mu_f$ by the equation (1) in the manuscript. We define two distributions B and C over $\delta = 0$ and $-\infty$ such that $B(0) = p{\cdot}\alpha$ and $C(0) = \gamma{\cdot}\mu_f$. $B(-\infty) := 1-B(0)$ and $C(-\infty) := 1 - C(0)$ are called *noise probabilities*. The *divergence* of the feature f is defined by the Kullback-Leibler (KL) divergence between B and C.

When more than one ion types are considered, we define two distributions B and C over ion types $\delta \in \Delta$ and $-\infty$ such that $B(\delta) = p \cdot \alpha_\delta$ and $C(\delta) = \gamma_\delta \cdot \mu_f^\delta$. The noise probabilities for B and C are given by $1 - \sum_{\delta \in \Delta} B(\delta)$ and $1 - \sum_{\delta \in \Delta} C(\delta)$, respectively. The divergence of f is defined by the KL divergence between B and C. The features in the feature set F are ranked according to the divergences (the higher divergence, the higher rank) after the training of UniNovo.

Given a peak i, all features that the peak i satisfies are divided into different groups as follows: First, the linking features make one group. Second, the offset features make the second group (in the extended model described below, the offset features are again divided into different groups according to the combination of terminal T, base peak charge z_1, and support peak charge z_2.[5]) Then, per each group of features, we select the highest ranking feature for the set H. All features in H are assumed to be independent and thus the $\mathcal{FPV}$ is obtained as above.

2.2.6 Generating *de novo* reconstructions

To generate *de novo* reconstructions, we first construct a *spectrum graph* [DAC+99]. Given a spectrum S of parent mass n from an unknown peptide P, the spectrum graph $G(V, E)$ is defined as a directed acyclic graph whose vertex set V consists of 0 (the source), n (the sink), and integers i such that $\mathcal{FPV}_i > 0$. Two vertices i and j are connected by an edge (i, j) if $j - i$ equals to the mass of an amino acid or the total mass of multiple amino acids (*a mass gap*). Any path from 0 (the source) to n (the sink) in a spectrum graph corresponds to a peptide (possibly containing mass gaps). We say that a vertex i is *correct* if $P_i = 1$ and an edge (i, j) is *correct* if both vertices i and j are correct. We also say that a path r is *correct* if all vertices in r are correct. The *length* of a reconstruction is defined by the total number of amino acids and mass gaps in the reconstruction.

To score a *de novo* reconstruction, we use an additive (i.e., the score of a path is the sum of scores of vertices of the path) log likelihood ratio scoring (similar to [DAC+99]). Given a vertex i, let $\mathcal{FPV}_i = x$. The likelihoods of the following two hypothesis for the outcome $\mathcal{FPV}_i = x$ are tested: a) the vertex i is correct and b)

[5]The rationale behind this selection is that two ions of the different terminus or charge states are likely to be weakly correlated each other.

the vertex i is incorrect. Let $Pr(P_i = 1|\mathcal{FPV}_i = x) = x$. Then, we have

$$\frac{\mathcal{L}(P_i = 1|\mathcal{FPV}_i = x)}{\mathcal{L}(P_i = 0|\mathcal{FPV}_i = x)} = \frac{Pr(\mathcal{FPV}_i = x|P_i = 1)}{Pr(\mathcal{FPV}_i = x|P_i = 0)} = \frac{x}{1-x} \cdot \frac{1-p}{p}. \tag{2.19}$$

The score of the vertex i with $\mathcal{FPV}_i = x$ is defined by $Score(i) := \left[\log \frac{x}{1-x} \cdot \frac{1-p}{p}\right]$ where $[\,\cdot\,]$ denotes the rounding to the nearest integer. Given a path r, the score of the path r is defined by $\sum_{i \in r} Score(i)$.

Since an additive scoring is used, top scoring reconstructions can be efficiently generated using a dynamic programming as in [DAC+99]. We did not exclude symmetric paths in the spectrum graph that usually correspond to incorrect reconstructions. Considering only the antisymmetric paths would further enhance the performance of UniNovo [CKT+01].

After generating the reconstructions, a probability that each reconstruction is correct (termed the *accuracy* of the reconstruction) is predicted, using Hunter's bound [Hun76]. To estimate the accuracy of a reconstruction (i.e., a probability that the reconstruction is correct), we first learn one more statistic from the training dataset $\mathcal{T}$: *EdgeAccuracy* of edges in the spectrum graph. Given an edge (i, j), $EdgeAccuracy(i, j)$ is an empirical probability of (i, j) being correct. More precisely, each edge (i, j) is characterized by the following quantities: $\mathcal{FPV}_i$, $\mathcal{FPV}_j$ (quantized into 10 levels), and the minimum amino acid number whose total mass equals to $j - i$. Call these quantities of an edge the *property* of the edge. From the training dataset $\mathcal{T}$, we obtain the empirical probability that an edge with a property is correct for all possible properties. Then, given an edge (i, j) (generated from a query spectrum), denote the learned empirical probability for the property of the edge by q. The $EdgeAccuracy(i, j)$ is given by $\min(q, \mathcal{FPV}_i, \mathcal{FPV}_j)$.[6]

The accuracy of a reconstruction is then derived from $\mathcal{FPV}_i$ of its vertices i and $EdgeAccuracy(i, j)$ of its edges (i, j) using an upper bound for the probability of a union proposed by Hunter, 1976 [Hun76]. Given a reconstruction $r = \{i_0, \cdots, i_l\}$ on the spectrum graph (of a spectrum S from an unknown peptide P), we consider a

[6] $\mathcal{FPV}_i$ estimates the probability that the vertex i is correct, and $EdgeAccuracy(i, j)$ estimates the probability both the vertices i and j are correct. To construct a probability space based on these estimates (see below), $EdgeAccuracy(i, j)$ is forced to be smaller than both $\mathcal{FPV}_i$ and $\mathcal{FPV}_j$.

probability space $(\Omega_r, \mathcal{F}_r, Pr_r)$ whose sample space Ω_r is given by

$$\Omega_r = \{P_i = x : i \in r,\ x = 0, 1\}. \tag{2.20}$$

The set of events $\mathcal{F}_s$ is composed of all subsets of Ω_r. Let $Pr_r(P_i = 1) = \mathcal{FPV}_i$, and $Pr_r(P_i = 1, P_{i'} = 1) = EdgeAccuracy(i, i')$ for i, i' in r. The probability we want to derive can be written as $Pr_r(\bigcap_{i \in r} P_i = 1)$.

To use Hunter's bound, we construct a complete graph[7] $\mathcal{E}$ whose vertex i represents the event $P_i = 1$ for $i \in r$. The weight of an edge (i, j) is defined by

$$\begin{aligned} w_{(i,j)} &= Pr_r(P_i = 1 \text{ or } P_j = 1) && (2.21)\\ &= Pr_r(P_i = 1) + Pr_r(P_j = 1) - Pr_r(P_i = 1, P_j = 1) && (2.22)\\ &= \mathcal{FPV}_i + \mathcal{FPV}_j - EdgeAccuracy(i, j). && (2.23) \end{aligned}$$

Hunter's bound gives us the following bound:

$$\begin{aligned} Pr_r(\bigcap_{i \in r} P_i = 1)) &\geq \sum_{i \in r} Pr_r(P_i = 1) - \sum_{(i,j) \in T_{\mathcal{E}}} w_{(i,j)} && (2.24)\\ = \sum_{i \in r} \mathcal{FPV}_i &- \sum_{(i,j) \in T_{\mathcal{E}}} (\mathcal{FPV}_i + \mathcal{FPV}_j - EdgeAccuracy(i, j)). && (2.25) \end{aligned}$$

where $T_{\mathcal{E}}$ is the minimum spanning tree on the graph $\mathcal{E}$. The expression in (2.25) defines the accuracy of the reconstruction r.

2.2.7 How to extend UniNovo algorithm for the realistic model

Here we describe how to extend the algorithm for the realistic model. We first introduce the realistic model we used and second explain the modified algorithm.

Changes in the model/definitions

In practice, UniNovo considers the model in which

- the mass tolerances of MS1 and MS2 are given by users

[7] An undirected graph in which every pair of distinct vertices is connected by a unique edge.

- the mono-isotopic masses of amino acids are used (e.g., the mass of Gly is 57.021464), and the m/z positions of peaks are real numbers (when MS1 tolerance is smaller than 0.1 Da; otherwise, integer amino acid masses and m/z values are used)
- all spectra are divided into 5 groups according to their parent mass ranges (see Table 2.2 (a))
- the intensities of peaks are divided into 11 levels (see $I(S_i)$ in Table 2.2 (b))
- only 150 peaks with high intensities are considered per a spectrum
- both N- and C- terminal ions of any charge values up to 4 are considered (e.g., b, y, b^2 ions)

The peak i in a spectrum S refers to the peak whose m/z value equals to i within MS1 mass tolerance.[8] The raw intensity of the peak i is denoted by S_i and the intensity level of it is by $I(S_i)$. If multiple such peaks are present, simply pick the highest intensity one. UniNovo learns/applies all parameters (α, β, μ, and ν) separately for different groups of spectra. Also, the parameters are learned separately for the fragmentation sites corresponding to the enzyme specific amino acids (e.g., C-terminal K or R for tryptic peptides). Except those amino acids, the current version of UniNovo does not take amino acid specific information (e.g., the different propensities of amino acids or the effect of proline on fragmentations) into account. Many studies have reported that different amino acids alter the fragmentation characteristics of MS/MS spectra [WTSB00, HTT+05]. By considering amino acids differently as in PepNovo, the performance of UniNovo could be improved; however, the number of annotated spectra necessary for the training of UniNovo should be also increased by orders of magnitude to avoid overfitting, which may weaken the universal property of UniNovo. A possible idea to mitigate such a negative effect may be to cluster amino acids into a small number of groups (e.g., basic and non-basic groups) and to train the parameters separately for each group, which will be included in our future work.

[8] Even if a different mass tolerance is used, no fundamental change is necessary for UniNovo algorithm. We only need to redefine what the definition of peak i in a spectrum is.

The definition of a feature f is changed so that it can accommodate the changed model. Before we define a feature f, define the *intensity ratio function* $R(u,t)$, a function from two real numbers u,t to an integer, as in Table 2.2 (c). A feature $f=(t,x,r,T,z_1,z_2)$ is now a vector with 6 elements (instead of a single integer in the manuscript): intensity t, mass offset x, intensity ratio r, terminal T, base peak charge z_1, and support peak charge z_2. The mass offset x represents a mass gain/loss, and T shows if the feature represents the relation between the ions of the same terminal ($T=0$) or not ($T=1$). Given a spectrum S from a peptide of mass n,[9] a peak i in S is said to *satisfy* $f=(t,x,r,T,z_1,z_2)$ if $I(S_i)=t$ and there exists another peak j such that $R(S_i,S_j)=r$ where j is given by

$$j=\begin{cases}\frac{z_1\cdot(i-\epsilon)+x}{z_2}+\epsilon & \text{if } T=0\\ \frac{n-(z_1\cdot(i-\epsilon)+x)}{z_2}+\epsilon & \text{otherwise.}\end{cases}$$

where ϵ is the mass of a proton. With the new definition, a feature can characterize the m/z (by specifying x) and intensity relation (by specifying t and r) between two peaks of ion types of different terminus (by specifying T) and/or different charges (by specifying z_1 and z_2).

Iterative training/running for better ion type inference

The $\mathcal{FPV}$'s for different ion types can be used to assign a probability distribution ρ (over ion types in Δ and *noise*) to each peak such that $\rho(\delta)$ is the probability that the peak is a δ-ion peak and ρ(noise) is the probability that the peak is not a δ-ion peak for all $\delta\in\Delta$ (termed a *noisy* peak).

The distribution ρ is meaningful only when it is far from a uniform distribution. However, if the spectra in the training dataset $\mathcal{T}$ contain abundant noisy peaks or peaks of different ion types with similar characteristics, the distribution often has a uniform-like distribution. Thus, it can be more informative given a training dataset consisting of spectra containing few noisy peaks and peaks of different ion types with distinctive characteristics.

To obtain such a training dataset, we generate *processed PSMs* from PSMs in the (original) training dataset. Given an ion type set Δ and the distribution ρ of the

[9]The peptide mass n can be calculated from the parent mass of the spectrum.

peak i in a spectrum S, the *processed spectrum* S' is a spectrum satisfying

$$S'_i = \sum_{\delta \in \Delta} \rho(\delta) \cdot OFF(\delta) \tag{2.26}$$

for all i such that $S_i > 0$. Since the intensity of a peak in the processed spectrum S' is a weighted summation of the distribution ρ of the peak, it is likely that the peaks in S' are clustered according to the ion types of the peaks (see Figure 2.2). Denote the (original) training dataset as $\mathcal{T}_1$. The ion type set Δ and feature set F_1 are learned from $\mathcal{T}_1$. For each PSM (P, S) in $\mathcal{T}_1$, the processed spectrum S' is generated from S using features in F_1, yielding a *processed PSM* (P, S'). The resulting set of the processed PSMs is denoted as $\mathcal{T}_2$. Likewise, we repeat generating $\mathcal{T}_{i+1}$ using features in F_i learned from $\mathcal{T}_i$ for $i = 1, \cdots, 4$.[10] The feature sets $F_1, \cdots, F_4$ are kept by UniNovo.

An input spectrum undergoes the same iterative process. Denote an input spectrum as S_1. We generate the (processed) spectrum S_{i+1} from S_i using features in F_i learned from $\mathcal{T}_i$ for $i = 1, \cdots, 4$. The $\mathcal{FPV}$ is generated based on the distributions ρ after 5 iterations.

2.3 Results

2.3.1 Datasets

To benchmark UniNovo, we used 13 different datasets with diverse fragmentation methods (CID/ETD/HCD), digested with diverse proteases (trypsin, LysC, and AspN), and having diverse charge states (see Table 2.1 and below). We re-analyzed the spectral datasets (*original datasets*) from Albert Heck's and Joshua Coon's laboratories that were previously analyzed in [KMB+10], [SWC10], and [FAH+11]. The CID and ETD spectra in these original datasets were acquired in a hybrid linear ion trap/Orbitrap mass spectrometers (high MS1 resolution and low MS2 resolution). The HCD spectra have high MS1 and MS2 resolution.

All spectra in the original datasets were identified by MS-GFDB (ver. 01/06/2012) [KMB+10] at 1% peptide-level FDR without allowing any modification except

[10] After 5 iterations, no significant changes were observed in the resulting training dataset.

the carbamidomethylation of Cys (C+57) as a fixed modification. Out of all identified spectra, we selected 1,000 spectra (or pairs of spectra) from distinct peptides randomly and formed the 13 datasets listed in Table 2.1. The unselected identified spectra (about 5,000-20,000 spectra depending on the type of spectra) were used for the training of UniNovo. The peptide contained in the training dataset were not contained in the 13 datasets. The detailed description of the each dataset is as follows:

CID, ETD, and CID/ETD datasets: CID, ETD, and CID/ETD datasets contain LTQ-Orbitrap spectra (Thermo Fisher Scientific) of trypsin digested peptides from the human HEK293 cell line generated in Albert Heck's laboratory (see [KMB$^+$10] for details). The original dataset described in [KMB$^+$10] contains the CID/ETD spectral pairs. To obtain the CID dataset, we took only CID spectra from the original dataset and identified them using MS-GFDB at 1% FDR. Out of the identified CID spectra, we randomly pick 1,000 doubly charged spectra that represent distinct tryptic peptides. ETD dataset was generated similarly, and it consists of 1,000 doubly and 1,000 triply charged ETD spectra of distinct tryptic peptides. CID/ETD dataset contains 1,000 pairs of doubly charged and 1,000 pairs of triply charged CID/ETD spectra of distinct tryptic peptides.

CIDL, CIDA, ETDL, and ETDA datasets: To benchmark UniNovo on spectra of non-tryptic peptides, we analyzed 4 spectral datasets generated in Joshua Coon's laboratory (see [SWC10] for details). From yeast protein samples, the authors in [SWC10] generated CID and ETD spectra of LysC or AspN digested peptides on a hybrid linear ion trap-orbitrap mass spectrometer (Thermo Fisher Scientific). From the identified CID spectra of LysC (AspN) digested peptides, we randomly pick 1,000 doubly charged spectra representing distinct peptides to generate CIDL (CIDA) dataset. In case of ETD spectra, we selected 1,000 charge 3 and 1,000 charge 4 spectra representing distinct LysC (AspN) digested peptides to generate ETDL (ETDA) dataset.

HCD dataset: To generate HCD dataset, we used HCD spectra reported by [FAH$^+$11]. The original spectra were acquired by LTQ-Orbitrap Velos (Thermo Fisher Scientific) using one of three different fragmentation methods (CID, ETD, and HCD) from trypsin digested peptides of HEK293 whole cell lysates. We took only the HCD spectra from the original dataset and identified them using MS-GFDB. Out

of all identified spectra, we randomly sampled 1,000 doubly charged and 1,000 triply charged spectra of distinct tryptic peptides.

2.3.2 Benchmarking UniNovo

We benchmarked UniNovo, PepNovo+ (ver. 3.1 beta) [Fra09], PEAKS (ver. 5.3, online) [MZH$^+$03], and pNovo (ver. 1.1) [CSY$^+$10] using the datasets in Table 2.1. For each tool, we generated N *de novo* reconstructions per each spectrum for N = 1, 5, and 20. We say that a spectrum is *correctly sequenced* if at least one of N reconstructions generated from the spectrum is correct. To evaluate the performance of each tool, the number of correctly sequenced spectra and the average length of correct reconstructions were measured for each tool.[11]

For UniNovo, the maximum number of mass gaps in a reconstruction was set to 2. UniNovo was tested for all datasets. For PepNovo+, also N top scoring reconstructions were generated per spectrum. PepNovo+ was used for CID2, CIDL2, CIDA2, HCD2, and HCD3 datasets. In case of PEAKS, we first generated 500 top scoring reconstructions per each spectrum. Then, for each reconstruction we converted amino acids with the local confidence lower than 30% into mass gaps. Such conversion is adopted because PEAKS generates reconstruction without mass gaps while UniNovo and PepNovo+ generate reconstructions with up to two mass gaps. In this procedure, multiple reconstructions without mass gaps were often converted into the same reconstruction with mass gaps. The score of a converted reconstruction is defined as the highest score of the reconstructions before conversion. Out of the converted reconstructions, N top high scoring (distinct) ones were chosen and used for further analysis. PEAKS was tested for all datasets except for HCD2 and HCD3 datasets. For pNovo, N top scoring reconstructions were generated per a spectrum.[12] Only HCD2 and HCD3 datasets were analyzed by pNovo.

We also indirectly compared UniNovo with MS-GFDB [KMB$^+$10] as both

[11]Since mass gaps are allowed for reconstructions, often multiple correct reconstructions were reported for a spectrum. To calculate the average length of correct reconstructions, only the top scoring correct reconstruction was counted per a spectrum.

[12]pNovo also generates reconstructions without mass gaps. However, the conversion of reconstructions as in PEAKS could not be applied to pNovo because pNovo does not report any local score.

tools were developed to analyze diverse types of spectra. We replaced the scoring function of UniNovo with that of MS-GFDB and generated reconstructions using the replaced scoring method. More precisely, the spectrum graph was generated by MS-GFDB per each spectrum, and the reconstructions were generated by UniNovo on that spectrum graph (instead of the spectrum graph generated by UniNovo). This generation method is specified by MS-GFDBScore. All experimental parameters for MS-GFDBScore were the same as for UniNovo.

Figure 2.3 shows the comparison results for different datasets. UniNovo found the largest number of correctly sequenced spectra among all the tested tools in most datasets. In particular, for ETD spectra, UniNovo reported significantly more correctly sequenced spectra than PEAKS. For example, in case of ETD2 or ETDL4 dataset, the number of correctly sequenced spectra was more than twice for UniNovo than for PEAKS.

For CID spectra, UniNovo and PepNovo+ showed similar results. When $N = 1$, UniNovo and PepNovo+ found about the same number of correctly sequenced spectra in CID2 and CIDL2 datasets, but UniNovo found about 35% more correctly sequenced spectra than PepNovo+ in CIDA2 dataset.

While trypsin and LysC digested peptides generate the spectra of similar fragmentation characteristics, AspN digested peptides generate spectra with distinct fragmentation propensities. UniNovo worked well with AspN digested peptides, but PepNovo+ showed suboptimal results for the spectra of AspN digested peptides.[13] The length of correct reconstructions for PepNovo+ was slightly longer than for UniNovo.

The results on HCD spectra also demonstrate that UniNovo finds the largest number of correctly sequenced spectra in general. The reconstructions reported by pNovo were, however, longer than those by UniNovo (and PepNovo+) by 2-3 amino acids. This suggests that UniNovo still has room for improvement for HCD spectra (e.g., introducing features better reflecting the high mass resolution and information from immonium or internal ions).

The results from UniNovo were superior to MS-GFDBScore in both terms

[13]Training of the parameters for the Bayesian network of PepNovo [FP05] for the CID spectra of AspN or LysC digested peptides would lead to better results; however, as mentioned above, the re-ranking models of PepNovo+ [Fra09], which are crucial for the suprior performance of PepNovo+ for CID tryptic spectra, cannot be readily trained.

of the number of correctly sequenced spectra and the average length of the correct reconstructions in all datasets.

For each dataset, we drew the Venn diagrams of the correctly sequenced spectra (Figure 2.4) to see the overlaps of the spectra between different tools. For all datasets, the overlaps between different tools increase as N grows, as expected. Relatively small overlaps are observed for ETD spectra (as compared to CID or HCD spectra). It indicates that UniNovo may have been using some valuable features of ETD spectra missed by PEAKS (and vice versa) and suggests that combining UniNovo and PEAKS results may potentially lead to a promising *de novo* sequencing approach.

While the above results measure the sequence level accuracy, they do not directly show the amino acid level precision or recall. To measure the amino acid level precision and recall, the top scoring reconstruction was generated per spectrum for each tool (i.e., $N = 1$). For this experiment, MS-GFDB was not tested, and the reconstructions of PEAKS were not converted using the local confidence. From the generated reconstructions, the number of (predicted) fragmentation sites and the number of correct fragmentation sites are counted. Also, since the spectra are annotated, we can count the number of all fragmentation sites in test sets. The precision and recall are defined by

$$\text{precision} = \frac{\text{\# correct fragmentation sites}}{\text{\# predicted fragmentation sites}} \tag{2.27}$$

$$\text{recall} = \frac{\text{\# correct fragmentation sites}}{\text{\# all fragmentation sites in test sets}}. \tag{2.28}$$

Figure 2.5 shows the precision and recall values of the tested tools for different datasets. For all datasets, UniNovo showed the highest precision value. But the recall values of UniNovo tended to be lower than others in particular for CID spectra. For ETD2 and ETDL4 datasets, UniNovo had higher precision and recall than PEAKS. These observations are consistent with the sequence level results above; higher precision of UniNovo resulted in more accurate reconstructions, and lower recall resulted in shorter reconstructions.

Both the sequence level and amino acid level results suggest that specific types of spectra are more suitable for *de novo* sequencing than others. For instance, in

general, HCD spectra generated more accurate and longer reconstructions (or higher precision and recall in amino acid level) than ETD spectra.

2.3.3 Evaluation of the spectrum graph

For further evaluation of the scoring function (i.e., spectrum graph) of UniNovo for different spectrum types, we also compared the spectrum graphs from UniNovo, PepNovo, and MS-GFDB for CID2 dataset.

To evaluate the spectrum graph of UniNovo for different spectrum types, we plotted ROC (*Receiver Operating Characteristic*) curves of vertices (i.e., plausible fragmentation sites) in the spectrum graphs from each dataset. Given a spectrum graph, we first ranked all vertices in such a way that the xth highest scoring vertex has the rank x. Then we chose 20 top ranking vertices (excluding source and sink vertices), and calculated true positive rates and false positive rates at various rank thresholds. For a rank threshold x, the true positive rate was calculated by # of correct vertices of rank less than x divided by # of correct vertices, and the false positive rate was by # of incorrect vertices of rank less than x divided by # of incorrect vertices.[14]

Figure 2.7(a) shows the ROC curves for the datasets in Table 2.1, except the ones of spectral pairs. For CID2 dataset, the ROC curves of spectrum graphs generated by MS-GFDB and PepNovo+ were also drawn. The ROC curve of UniNovo for CID2 dataset (blue circled line) is significantly better than those of PepNovo+ and MS-GFDB (black and black dashed lines). For instance, at the false positive rate of 0.1, the true positive rate of UniNovo was 0.7 while both MS-GFDB's and PepNovo+'s were about 0.4. As the ROC curves suggest, HCD2 (ETDL4) datasets represents the most (the least) suitable datasets for *de novo* sequencing. Other datasets can be ranked as: HCD2 (the best)$\rightarrow$CID2 $\rightarrow$ ETD2 $\rightarrow$ CIDL2 $\approx$ CIDA2 $\approx$ ETDL3 $\approx$ ETDA3 $\approx$ ETDA4$\approx$ HCD3 $\rightarrow$ ETD3 $\approx$ ETDL4 (the worst).

The above ROC curve evaluates the sensitivity/specificity of the scoring functions with 20 highest ranking vertices in the spectrum graph. However, if only few of the 20 vertices are correct - in other words, most fragmentation sites are not selected

[14]The error tolerance for vertices was set to 0.5 Da except for HCD2 and HCD3 datasets. For HCD2 and HCD3 datasets, the error tolerance was set to 20 ppm.

for 20 vertices - such an evaluation may be pointless. Thus, we also measured the fraction of all fragmentation sites that are actually included in the correct vertices of rank less than x (i.e., the number of correct vertices of rank less than x divided by the number of all fragmentation sites). The same measurement was done for CID2 dataset by PepNovo+ and MS-GFDB. Figure 2.7 (b) shows that UniNovo (blue circled line) correctly detected 20% and 40% more fragmentation sites within top 20 vertices than PepNovo+ and MS-GFDB (black and black dashed lines), respectively. Together with the ROC comparison, one can deduce that UniNovo detects more fragmentation sites and scores them more specifically than PepNovo+ or MS-GFDB. Also one can infer that the good performance of PepNovo+ shown in Figure 2.3 is obtained by reranking of the reconstructions using the sequence specific features. From Figure 2.7 (b), we can evaluate each dataset in terms of the fraction of correctly predicted fragmentation sites as: HCD2 (the best)→CID2 → CIDL2 → CIDA2 ≈ ETD2 ≈ ETDL3 ≈ ETDA3 → HCD3 → ETDA4 → ETD3 → ETDL4 (the worst).

2.3.4 *De novo* sequencing of paired spectra

UniNovo also can be used to sequence paired spectra (e.g., CID/ETD spectral pairs). Given multiple spectra from the same precursor ion, UniNovo first generates a spectrum graph from each of the spectra and next merges the spectrum graphs into a combined spectrum graph, on which the reconstructions are generated.

Given multiple spectrum graphs $G^1, \cdots, G^n$, we first define a merged spectrum graph G as follows: We need to define the vertices along with their scores in G. The vertices of G are given by the union of vertices of the input spectrum graphs. The score of a vertex i in G is given by $\sum_{k=1}^{n} G_i^k$. To calculate the accuracy of a reconstruction r on the merged graph G, we also need to redefine $\mathcal{FPV}$ and $EdgeAccuracy$ of G. For each the input spectrum graph G^k, $\mathcal{FPV}_i$ and $EdgeAccuracy(i, j)$ are defined for each i and j. The $\mathcal{FPV}_i$ ($EdgeAccuracy(i, j)$) of G is simply defined as the maximum value of these $\mathcal{FPV}_i$ ($EdgeAccuracy(i, j)$) of the input spectrum graphs.

To benchmark UniNovo in *de novo* sequencing of paired spectra, CID/ETD2 and CID/ETD3 datasets were analyzed by UniNovo. From CID/ETD2 dataset, two additional datasets were generated: CID/etd2 and cid/ETD2 datasets.

CID/etd2 dataset was formed by taking only CID spectra, and cid/ETD2 dataset by taking only ETD spectra in CID/ETD2 dataset. CID/etd3 and cid/ ETD3 datasets were generated similarly. For each dataset, we generated N =1, 5, and 20 top scoring reconstructions.

The results are shown in Figure 2.8. When precursor ions were doubly charged, the performance boost from the paired spectra was very modest. For $N = 1, 5$, and 20, UniNovo reported 5% more correctly sequenced spectral pairs in CID/ETD2 datasets than in CID/etd2 dataset. The average length of correct reconstructions for CID/ETD2 dataset was slightly longer than for CID/etd2 dataset.

In contrast, for triply charged spectra, the use of paired spectra was highly beneficial for generating more accurate reconstructions. For example, when $N = 1$, UniNovo reported 100% and 50% more correctly sequenced spectral pairs in CID/ETD3 dataset than in CID/etd3 and cid/ETD3 datasets, respectively. The length of correct reconstructions typically increases by 1-2 amino acids by using the CID/ETD paired spectra.

2.3.5 *De novo* sequencing with quality filtering

Given a set of reconstructions generated from a spectrum, UniNovo estimates the probability that at least one reconstruction in the set is correct (i.e., a probability that the spectrum is correctly sequenced) based on the accuracies of reconstructions as follows: Given a set of reconstructions $R = \{r_1, \cdots, r_N\}$, we define a variable D_i for $i = 1, \cdots, N$ as

$$D_i = \begin{cases} 1 & \text{if } r_i \text{ is correct} \\ 0 & \text{otherwise.} \end{cases} \tag{2.29}$$

We consider a probability space $(\Omega_R, \mathcal{F}_R, Pr_R)$ whose sample space Ω_R is defined by

$$\Omega_R = \{D_i = x : i = 1, \cdots, N,\ x = 0, 1\}. \tag{2.30}$$

The set of events $\mathcal{F}_R$ is composed of all subsets of Ω_R. Given two sequences r_i and r_j, we define $r_{i,j}$ as a reconstruction whose vertices are the union of those of r_i and r_j. For example, $r_1 = \{1, 2, 4, 5\}$ and $r_2 = \{1, 3, 4, 5\}$, $r_{1,2} = \{1, 2, 3, 4, 5\}$.

Denote the accuracy of a reconstruction r by $Acc(r)$. Let $Pr_R(D_i = 1) := Acc(r_i)$ and $Pr_R(D_i = 1, D_j = 1) := Acc(r_{i,j})$ for $i, j = 1, \cdots, N$. We assume that a sequence of (Bernoulli) random variables $D_1, D_2, \cdots, D_N$ forms a Markov chain.[15] The probability we want to estimate can be denoted by $Pr_R(\bigcup_{i=1}^{N} D_i = 1)$. Since $Pr_R(D_i = 1 \cup D_j = 1) = Acc(r_i) + Acc(r_j) - Acc(r_{i,j})$, we obtain

$$Pr_R(\bigcup_{i=1}^{N} D_i = 1) \tag{2.31}$$

$$= 1 - Pr_R(\bigcap_{i=1}^{N} D_i = 0) \tag{2.32}$$

$$= 1 - Pr_R(D_1 = 0) Pr_R(D_2 = 0 | D_1 = 0) \cdots Pr_R(D_N = 0 | D_{N-1} = 0, \cdots, D_1 = 0) \tag{2.33}$$

$$= 1 - Pr_R(D_1 = 0) \prod_{i=2}^{N} Pr_R(D_i = 0 | D_{i-1} = 0) \tag{2.34}$$

$$= 1 - Pr_R(D_1 = 0) \prod_{i=2}^{N} \frac{Pr_R(D_i = 0, D_{i-1} = 0)}{Pr_R(D_{i-1} = 0)} \tag{2.35}$$

$$= 1 - Pr_R(D_1 = 0) \prod_{i=2}^{N} \frac{1 - Pr_R(D_i = 1 \cup D_{i-1} = 1)}{Pr_R(D_{i-1} = 0)} \tag{2.36}$$

$$= 1 - (1 - Acc(r_1)) \prod_{i=2}^{N} \frac{1 - (Acc(r_i) + Acc(r_{i-1}) - Acc(r_{i,i-1}))}{1 - Acc(r_{i-1})} \tag{2.37}$$

where the equality between (2.33) and (2.34) is obtained from the Markov chain assumption. The right hand side of (2.37) defines the *set accuracy* of R[16], denoted by $SetAccuracy(R)$.

When the parameter N is set, one may want to choose N reconstructions with the highest accuracies to maximize the set accuracy. However, such a selection often results in a set of short reconstructions (because short reconstructions have relatively high accuracies). Since short reconstructions are not very useful in many cases (e.g.,

[15] This assumption is reasonable if two adjacent reconstructions in R are similar each other whereas other reconstructions are relatively dissimilar, which is often the case since reconstructions in R are sorted in the ascending order of their scores (see below).

[16] When multiple *de novo* reconstructions are reported, it is important to guarantee that one of them is correct.

in follow-up homology searches), UniNovo uses a greedy algorithm to select long and accurate reconstructions. Given parameters $SetAccuracyThreshold > 0$ and N, UniNovo tries to construct a reconstruction set R such that $SetAccracy(R) \geq SetAccuracyThreshold$ and $|R| \leq N$ by selecting both accurate and long reconstructions (long reconstructions are not accurate, in general). First UniNovo generates 100 high-scoring reconstructions (the *candidate* reconstruction set). The reconstructions in the candidate reconstruction set are sorted by their scores in descending order. Denote the sorted list of reconstructions as $C = \{r_1, r_2, \cdots, r_{100}\}$. A set of reconstructions R is initialized as an empty set, and an integer $MaxLength$ is initialized as one plus the length of the longest reconstruction in C. The reconstructions in C whose length are less than $MaxLength$ are added to R sequentially, starting from r_1. When $|R| = N$ or all reconstructions shorter than $MaxLength$ are added to R, $SetAccuracy(R)$ is calculated. If $SetAccuracy(R) \geq SetAccuracyThreshold$, UniNovo outputs R. Otherwise, $MaxLength$ is decreased by 1, R is again initialized as an empty set, and the above procedure is repeated until $MaxLength = 5$. If no output is generated when $MaxLength = 5$, the input spectrum is declared as a low quality spectrum and is filtered out.

We set $SetThreshold = 0.8$ and reanalyzed the datasets in Table 2.1. The maximum number of mass gaps per each reconstruction was set to 10. For each dataset, we measured the number of unfiltered spectra (termed *qualified* spectra) and the percentage of qualified spectra that were correctly sequenced (which is expected to be 80% since $SetAccuracyThreshold = 0.8$). The average length of correct reconstructions was also measured.

The results are given in Figure 2.6. For all datasets, the number of qualified spectra increases sharply as the number of reconstructions N grows (Figure 2.6 (a)). For example, UniNovo reported only few qualified spectra (less than 5) from CIDA2 dataset when $N = 1$. When $N = 20$, it reported more than 900 qualified spectra from the same dataset. In contrast to the dramatic changes in the number of qualified spectra, the percentage of qualified spectra that were correctly sequenced hardly changed across the datasets and the values of N (Figure 2.6 (b)). As expected, the percentage was around 80% for all cases (including the datasets containing CID/ETD spectral pairs), which shows that the set accuracy reported by UniNovo is reliable.

Figure 2.6 (c) shows the average length of correct reconstructions. As N decreases, the average length also decreases. This is because shorter reconstructions (with higher accuracies) are chosen by UniNovo when N is small to achieve high set accuracy.

2.4 Conclusion

We presented a universal *de novo* sequencing tool UniNovo that works well for various types of spectra. UniNovo can be easily trained for different types of spectra using only thousands of PSMs that typically can be obtained from a single MS/MS run. The experimental results show that UniNovo generates accurate and long *de novo* reconstructions from spectra of CID, ETD, HCD, and CID/ETD fragmentation methods and spectra of trypsin, LysC, or AspN digested peptides. We also showed that UniNovo is better than or comparable to other state of the art tools.

As pointed out by [MJ11], *de novo* sequences not only are valuable for the analysis of the novel peptides that are not present in proteome databases but also can facilitate the homology-based database searches. Since the reconstructions reported by UniNovo contain mass gaps representing the total mass of multiple amino acids (termed *gapped peptides* [KBP09, JKBP11]), MS-BPM algorithm [NAP11] can be used for fast exact or homology searches (UniNovo⊕MS-BPM). MS-BPM enables searches against a sequence database using gapped peptides as queries. Currently MS-BPM takes gapped peptides generated by MS-GappedDictionary [JKBP11] (MS-GappedDictionary⊕MS-BPM). However, the reconstructions from UniNovo are usually longer than those from MS-GappedDictionary (8-9 vs. 5-6). Since the search time of MS-BPM strongly depends on the length of gapped peptides - the longer gapped peptides, the shorter search time - the running time of UniNovo⊕MS-BPM is smaller than MS-GappedDictionary⊕MS-BPM by an order of magnitude in a blind search against the IPI Human proteome database ver.3.87 [KDW$^+$04] (data not shown).

Acknowledgements

Chapter 2, in full, was published as "UniNovo : a universal tool for *de novo* peptide sequencing". K. Jeong, S. Kim, and P. A. Pevzner. *Bioinformatics*, doi:

10.1093/bioinformatics/btt338. The dissertation author was the primary author of this paper.

Table 2.1: Summary of the datasets used for benchmarking. Number of spectra (or spectral pairs) is 1,000 for each dataset. While UniNovo is applicable to all datasets, other tools are only applicable to (or optimized for) datasets marked by '*'. PEAKS was not tested for HCD datasets.

Dataset	CID2	CIDL2	CIDA2	ETD2	ETD3	ETDL3	ETDL4	ETDA3	ETDA4	HCD2	HCD3
Fragmentation	CID	CID	CID	ETD	ETD	ETD	ETD	ETD	ETD	HCD	HCD
Charge	2	2	2	2	3	3	4	3	4	2	3
Enzyme	Tryp	LysC	AspN	Tryp	Tryp	LysC	LysC	AspN	AspN	Tryp	Tryp
Avg. pep. length	12.6	11.4	12.3	12.5	16.4	12.5	18.7	12.8	18.9	10.5	14.5
UniNovo		*	*	*	*	*	*	*	*	*	*
PepNovo+	*	*	*	N/A	N/A	N/A	N/A	N/A	N/A	*	*
PEAKS	*	*	*	*	*	*	*	*	*	*	*
pNovo	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	*	*

Dataset	CID/ETD2	CID/ETD3
Fragmentation	CID/ETD	CID/ETD
Charge	2	3
Enzyme	Tryp	Tryp
Avg. pep. length	12.3	17.1
UniNovo	*	*
PepNovo+	N/A	N/A
PEAKS	N/A	N/A
pNovo	N/A	N/A

Table 2.2: Partitioning of spectra and peak intensities. (a) partitioning of the spectra by their parent mass. 121.6 is the average amino acid mass. (b) the intensity level of a peak i in a spectrum S, denoted by $I(S_i)$. The intensity level of a peak is decided by its intensity rank (the ith highest intensity peak = rank i). (c) Definition of the intensity ratio function $R : \mathbb{R} \times \mathbb{R} \to \mathbb{Z}$. This function is used to define a feature.

(a)

# Partition	Precursor mass ($\times$121.6)
1	< 9
2	9-13
3	13-17
4	17-20
5	> 20

(b)

$I(S_i)$	Intensity rank of a peak i
10	1-10
9	11-20
8	21-30
7	31-40
6	41-50
5	51-60
4	61-70
3	71-80
2	81-90
1	91-150
0	$\geq$150

(c)

$R(u,t)$	u/t
$-\infty$	∞
-4	5-∞
-3	2.5-5
-2	1.7-2.5
-1	1.3-1.7
0	1.0-1.3
1	0.8-1.0
2	0.6-0.8
3	0.4-0.6
4	0.2-0.4
5	0.0-0.2

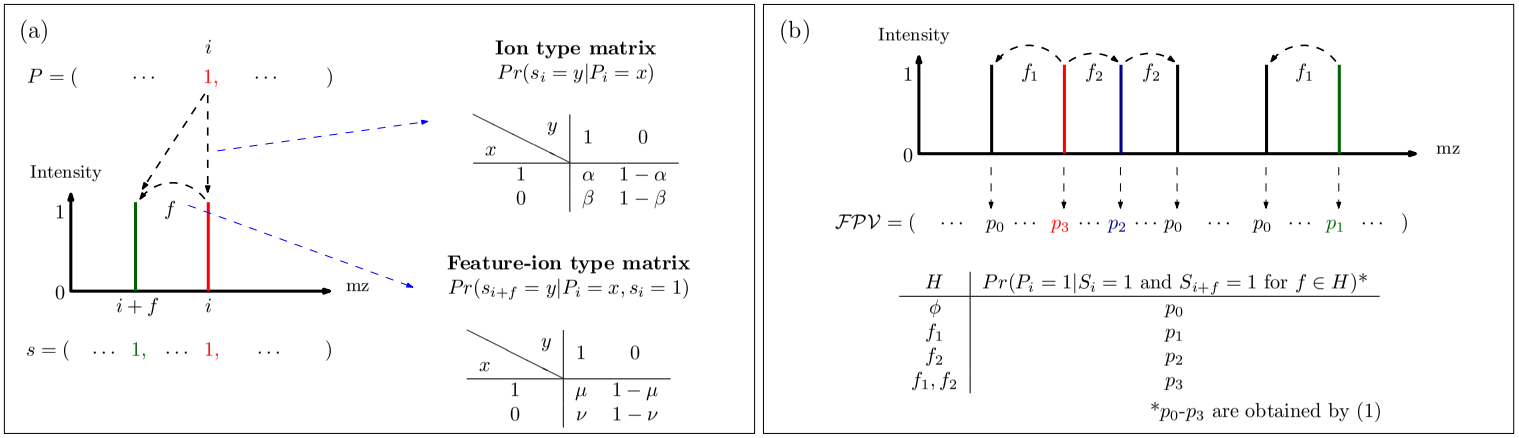
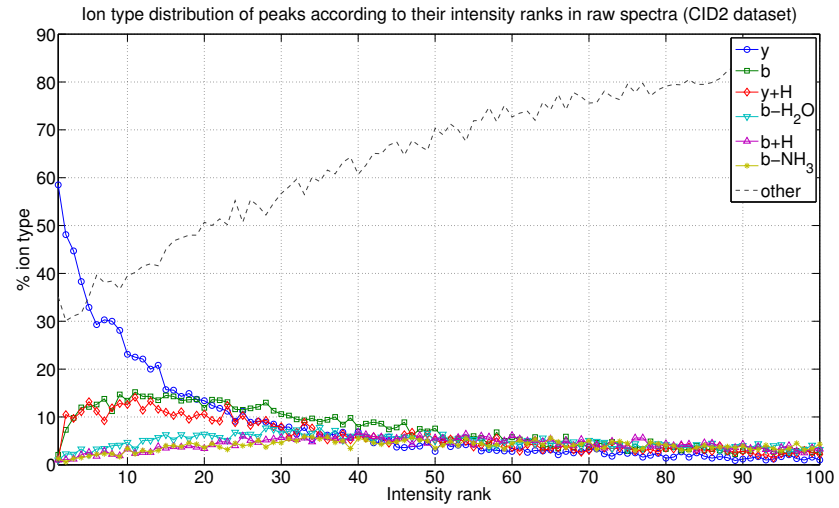
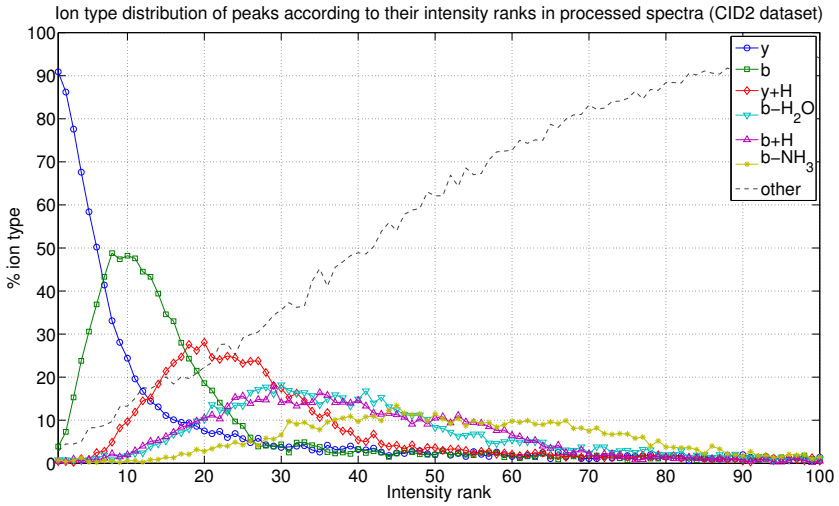


Figure 2.1: (a) The generation of a partial-spectrum s for P_i. One ion type $\delta = 0$ and one feature f are considered. The probability that $s_i = 1$ is given by α if $P_i = 1$ or by β otherwise. When $s_i = 1$, the probability that $s_{i+f} = 1$ (i.e., the peak i satisfies f) is given by μ if $P_i = 1$ or by ν otherwise. The spectrum is generated by taking elementwise OR operation for generated partial-spectra for all elements of P.
(b) The calculation of the fragmentation probability vector $\mathcal{FPV}$ from a spectrum S (without knowing the peptide P that generated S). We consider one ion type $\delta = 0$ and two features f_1 and f_2. The events "a peak satisfies f_1" and "a peak satisfies f_2" are assumed to be independent. To derive $\mathcal{FPV}_i$, first we examine which features the peak i satisfies in the spectrum S. Denote the features the peak i satisfies by H. Second, given H, we calculate the probability that $P_i = 1$ (using the probabilities given in ion type matrix and feature-ion type matrix - see the equation (2.15)).



(a)



(b)

Figure 2.2: Ion type distribution of peaks according to their intensity ranks for raw (a) and processed (after 5 iterations) spectra (b). CID2 dataset was used. A peak with the intensity rank i is the ith highest intensity peak in the spectrum. In raw spectra, different ion types are spread over the intensity ranks of peaks. Even in case of the highest intensity peaks, only 60% of them are y-ion peaks. The ion types in processed spectra are well clustered according to the intensity ranks of peaks. For example, 90% of the highest intensity peaks are y-ion peaks in processed spectra.

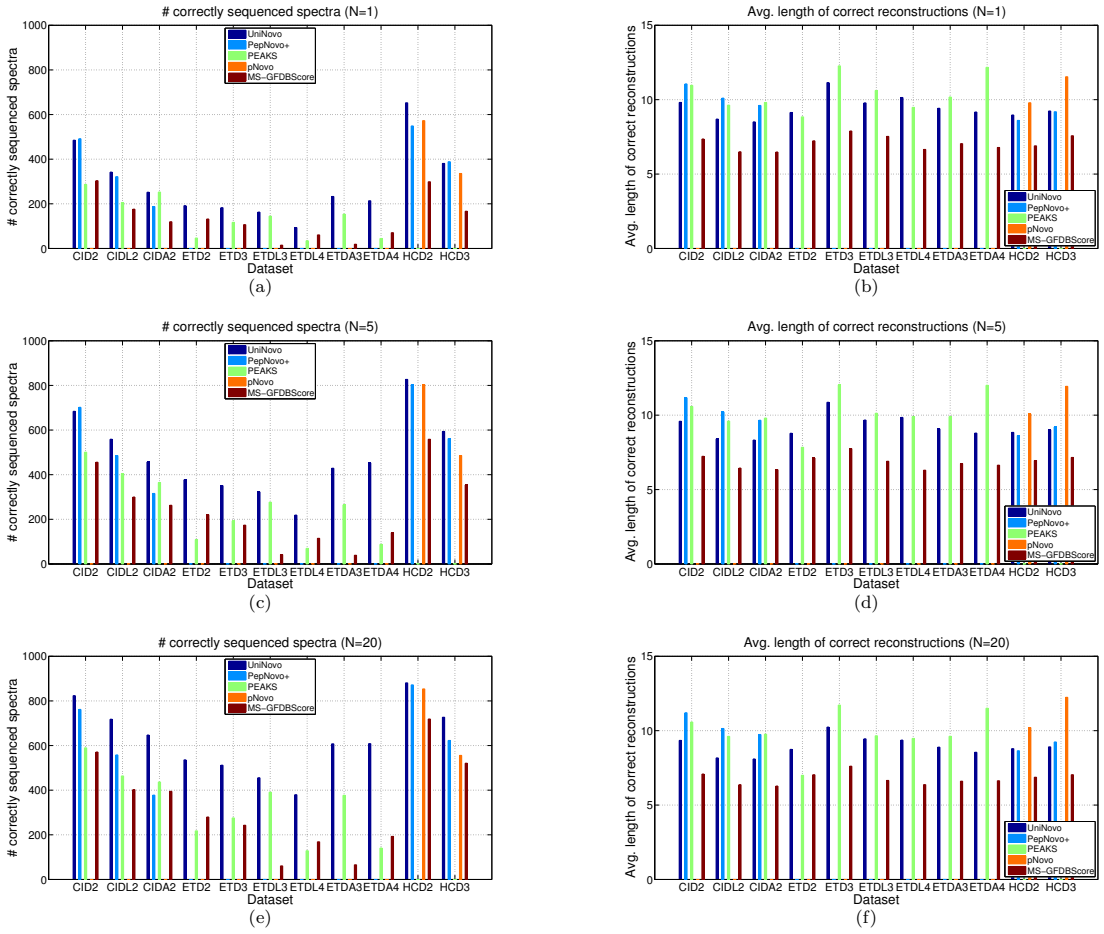


Figure 2.3: Comparison of *de novo* sequencing tools (as well as a database search tool MS-GFDB [KMB+10] tweaked for *de novo* sequencing). Per each spectrum, N top scoring reconstructions were generated by UniNovo, PepNovo+ [FP05, Fra09], PEAKS [MZH+03], pNovo [CSY+10], and MS-GFDBScore. MS-GFDBScore provides UniNovo with MS-GFDB's scoring function. The number of reported reconstructions per a spectrum (N) is set to 1, 5, and 20. A reconstruction is correct if all the fragmentation sites of the reconstruction are correct, and a spectrum is classified as correctly sequenced if at least one of the reconstructions generated from the spectrum is correct. Figures on the left side ((a), (c), and (e)) show the number of correctly sequenced spectra in each dataset, and figures on the right side ((b), (d), and (f)) show the average length of the correct reconstructions.

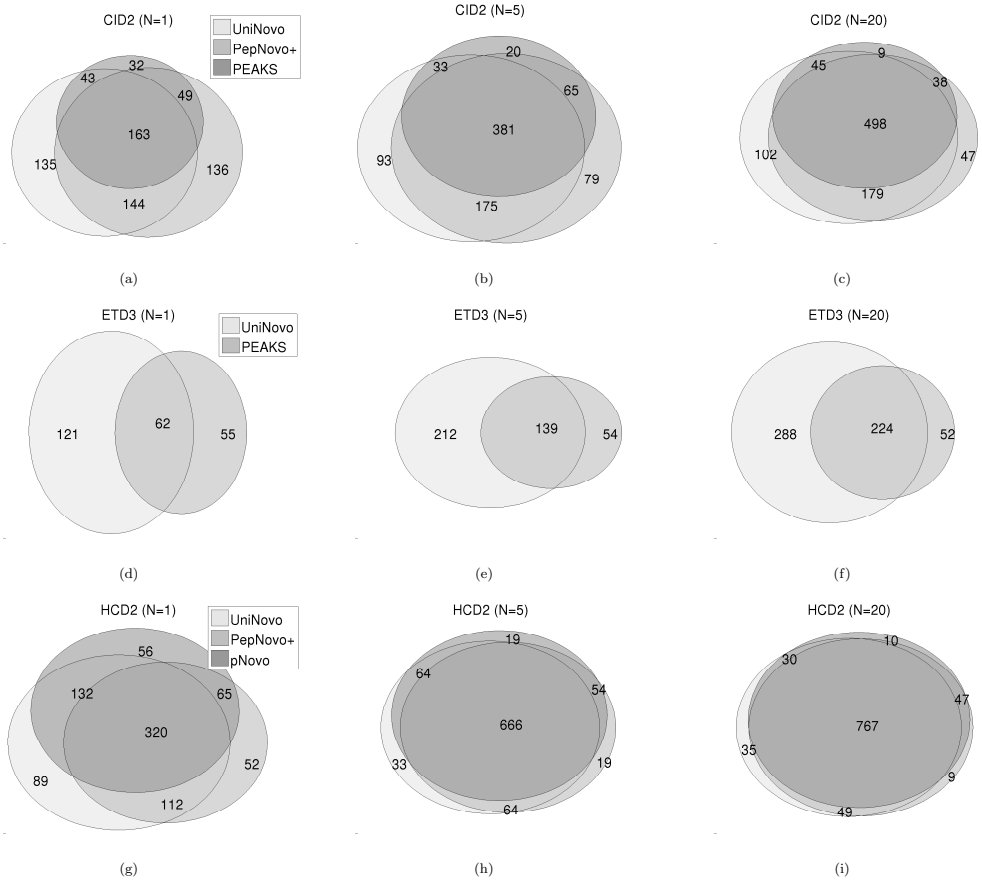


Figure 2.4: The Venn diagrams of the correctly sequenced spectra for CID2 (a)-(c), ETD3 (d)-(f), and HCD2 (g)-(i) datasets. For all datasets, the overlaps between different tools increase as N grows, as expected. Relatively small overlaps are observed for ETD spectra when compared to CID or HCD spectra.

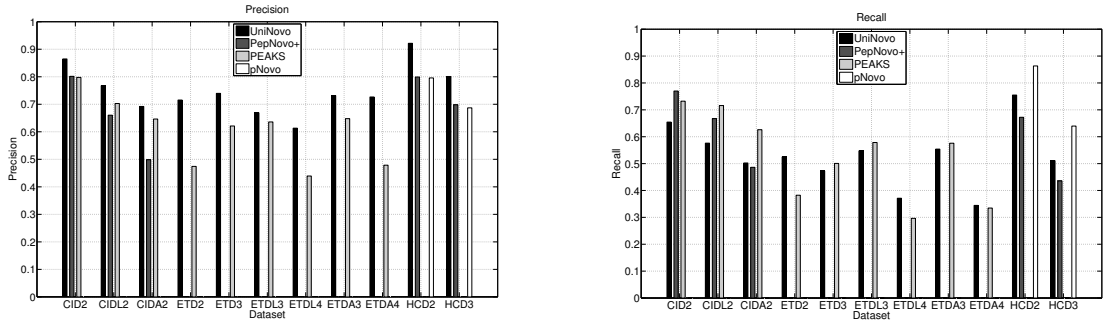
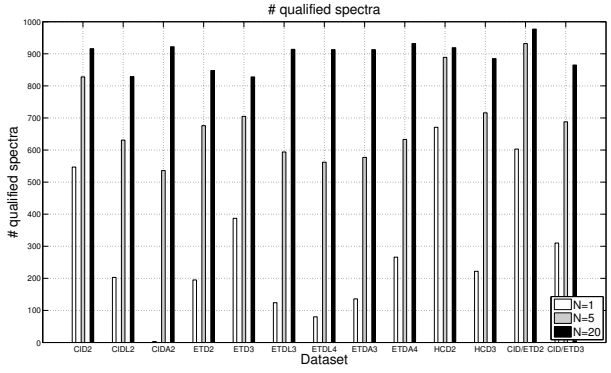
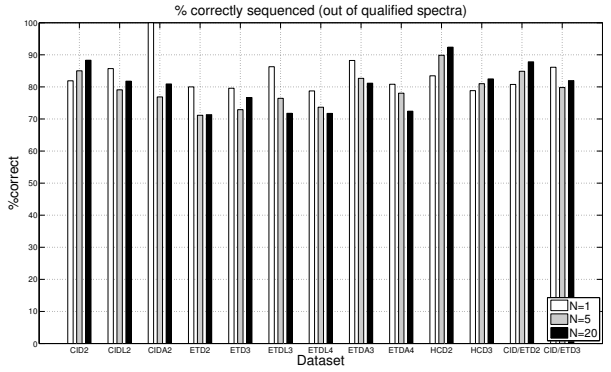


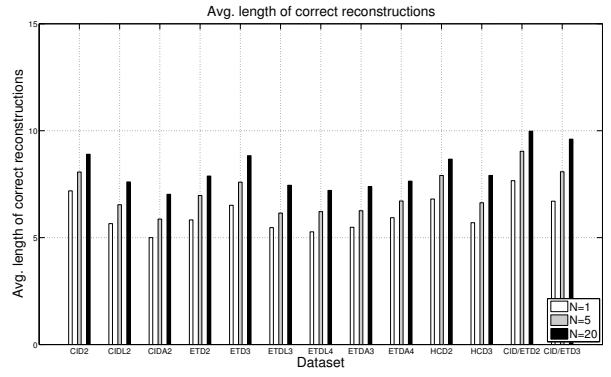
Figure 2.5: Comparison of *de novo* sequencing tools in terms of amino acid level precision (a) and recall (b). The definitions of precision and recall are given in (2.27) and (2.28), respectively.



(a)

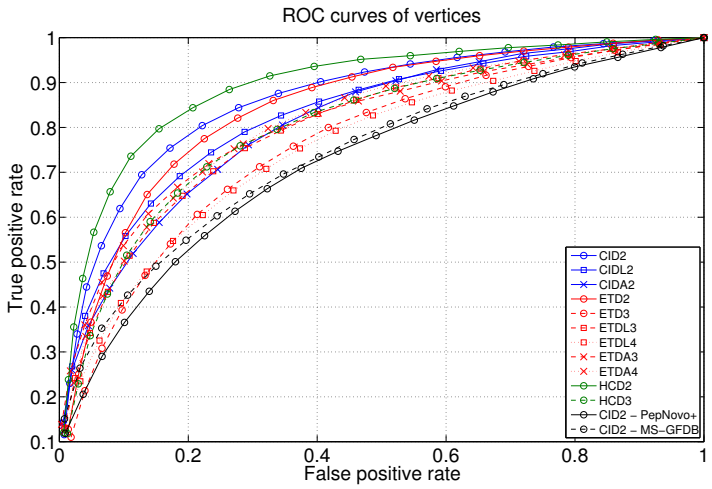


(b)

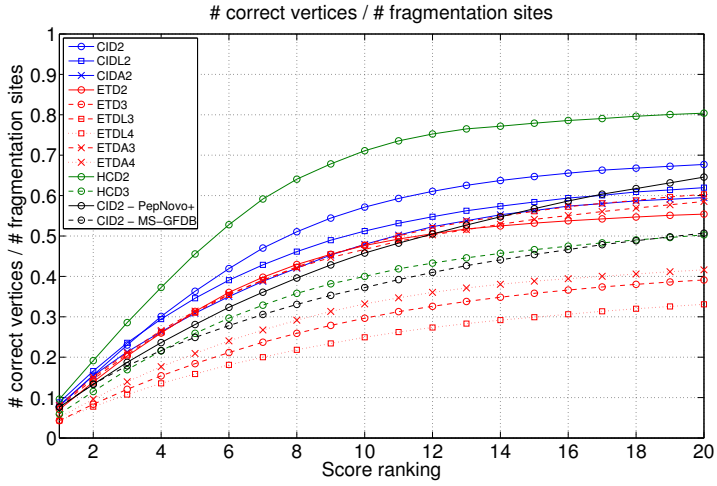


(c)

Figure 2.6: *De novo* sequencing with qualify filtering of spectra. Given a spectrum, if the parameter $SetAccuracyThreshold$ is set, UniNovo attempts to achieve set accuracy (an estimated probability of the spectrum being correctly sequenced) exceeding $SetAccuracyThreshold$. If it fails to generate such a set, the spectrum is filtered out. An unfiltered spectrum is called a *qualified spectrum*. We set $SetAccuracyThreshold = 0.8$. (a) the number of qualified spectrum, (b) the percentage of qualified spectra that were correctly sequenced, (c) the average length of correct reconstructions.

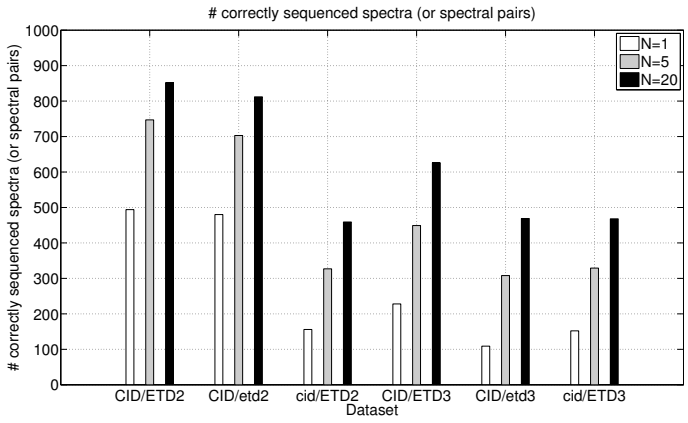


(a)

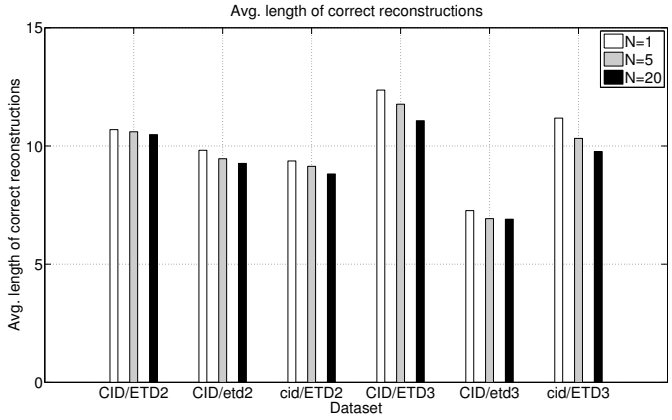


(b)

Figure 2.7: (a) ROC curves of vertices (i.e., plausible fragmentation sites) in the spectrum graphs. Per each spectrum graph, the vertices are ranked by their scores so that the xth highest scoring vertex has the rank x. We took 20 highest ranking vertices per each spectrum graph, and calculated the true positive rate and the false positive rate. Given a rank threshold x, the true (false) positive rate is given by # of correct (incorrect) vertices of rank less than x divided by # of correct (incorrect) vertices. Using UniNovo, ROC curves for the datasets in Table 2.1 (except the ones of spectral pairs) were generated. We also generated ROC curves using PepNovo+ (black line) and MS-GFDB (black dashed line) for CID2 dataset.
(b) The fraction of correctly predicted fragmentation sites. Given a rank threshold x, we measured what fraction of all fragmentation sites are included in the correct vertices of rank less than x.



(a)



(b)

Figure 2.8: *De novo* sequencing of paired spectra. CID/ETD spectral pairs were analyzed by UniNovo (in CID/ETD2 and CID/ETD3 datasets). To see if the spectral pairs are beneficial for *de novo* sequencing, CID/etd2 (cid/ETD2) dataset was generated from CID/ETD2 dataset by collecting only CID (ETD) spectra in CID/ETD2 dataset. Likewise, CID/etd3 and cid/ETD3 datasets were generated from CID/ETD3 dataset. (a) the number of correctly sequenced spectra (or spectral pairs), (b) the average length of correct reconstructions for each dataset. The spectral pairs resulted in more accurate and longer reconstructions, in particular for triply charged spectral pairs.

Chapter 3

Gapped Spectral Dictionaries and Their Applications for Database Searches of Tandem Mass Spectra

3.1 Introduction

Most peptide identification tools are rather slow since they match every tandem mass (MS/MS) spectrum against all peptides in a database (subject to constraints on the precursor mass, the enzyme specificity, and the number of missed cleavages). A faster approach would be to generate a full-length *de novo* reconstruction of a spectrum and to match the resulting peptide against a database. The fundamental algorithmic advantage of the latter approach is that one can pre-process the database (e.g., by constructing its suffix tree) so that matching becomes instantaneous. The only reason why most MS/MS database search tools still use the former approach is because full-length *de novo* peptide sequencing remains inaccurate. Even the most advanced *de novo* peptide sequencing tools [MZH$^+$03, FP05, Fra09] correctly reconstruct only 30 - 45% of the *complete* peptides identified in MS/MS database searches. After 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 MS/MS spectra (other reasons include limited understanding of fragmentation rules, co-eluted peptides, etc.). We

argue that regions with low information content should be represented as mass gaps (that represent two or more amino acids) and advocate use of gapped peptides as spectral interpretations.

Kim et al., 2009 [KGBP09] recently proposed to generate *multiple de novo* reconstructions (rather than a single one) and to match them against a database (MS-Dictionary approach). Since matching peptides against a pre-processed database is very fast, generating thousands of reconstructions still has advantages over the traditional approaches where spectra are matched against large databases. Given an MS/MS spectrum, MS-Dictionary generates the *Spectral Dictionary* [KGBP09] that contains all plausible *de novo* reconstructions of the spectrum (i.e., with scores exceeding a given threshold) and further matches them against a database. The running time of MS-Dictionary is almost independent of the database size making it a tool of choice for peptide identification in large databases [KGBP09].

Although MS-Dictionary was proved to be useful for peptides shorter than 15 amino acids (aa), it has limitations for longer peptides with large Spectral Dictionaries. For example, the size of the Spectral Dictionary for a typical 15 aa long peptide may exceed billion of peptides making it too large for MS/MS database search. We introduce MS-GappedDictionary that generates rather small *Gapped Spectral Dictionaries* (even for long peptides) thus addressing the key limitation of the Spectral Dictionaries. Gapped Spectral Dictionary is the set of *gapped peptides* (see [KBP09]) that are derived from the full-length peptides in the Spectral Dictionary. While the concept of a gapped peptide is not new [MZH$^+$03, FP05, SDW$^+$05, JT02, HJP$^+$01], constructing dictionaries of gapped peptides that account for all plausible *de novo* interpretations was not addressed before. Gapped peptides occupy a niche between accurate but short peptide sequence tags [MW94] and long but inaccurate full-length peptide reconstructions. The gapped peptides are both long and accurate making them well suited for *de novo*-based MS/MS database searches. In difference from short peptide sequence tags, a gapped peptide typically has a single match in a database reducing peptide identification to a single database look-up. For a typical 20-aa long peptide, the size of the Spectral Dictionary exceeds 10^{17}, while the size of the Gapped Spectral Dictionary is only $\approx 10^4$. Moreover, we show that even

smaller Gapped Spectral Dictionaries with only 20 - 100 peptides are sufficient for most applications. At the same time, gapped peptides are sufficiently long for efficient database matching. For example, for a spectrum of 15-aa long peptide, the average length[1] of gapped peptides in its Gapped Spectral Dictionary exceeds 9. For all practical purposes, (gapped) peptides of length 9 are as informative as (full-length) peptides of length 15 for matching databases (unless the database size approaches 20^9). Table 3.1 (a) shows the Gapped Spectral Dictionary of a spectrum of peptide LNRVSQGK shown in Figure 3.2 (a), consisting of 7 gapped peptides (as compared to its Spectral Dictionary consisting of 92 peptides). We describe an efficient algorithm for constructing the Gapped Spectral Dictionaries that also computes *coverage* of each gapped peptide, reflecting the portion of plausible *de novo* reconstructions represented by a gapped peptide (see below for the definition of coverage).

Recent proteogenomics studies highlighted the importance of MS/MS searches against the six-frame translation of genomes [YEM95, KMAM01, CBCC01, OWW$^+$02, TSN$^+$07, CPS$^+$08, BDK$^+$10, BGG$^+$08]. However, until recently, searches against the six-frame translations of large genomes were impractical even with the fastest MS/MS search tools, let alone with traditional tools like SEQUEST and Mascot. Although MS-Dictionary enabled searches in the six-frame translation of the human genome with 40X speed-up over InsPecT [KGBP09], it loses many peptide identifications (compared to InsPecT) because Spectral Dictionaries of long peptides have to be truncated (leading to truncating the correct peptides in some cases). Gapped Spectral Dictionaries remedy this shortcoming of Spectral Dictionaries and nearly double the number of identified peptides in the six-frame translation of the human genome (as compared to MS-Dictionary [KGBP09]).

Table 3.1 (b) illustrates how gapped peptides and their coverage can be utilized for constructing the *peptide sequence tags* [MW94]. Tanner et al., 2005 [TSF$^+$05] introduced *covering sets* of tags (set of tags containing at least one correct tag) and demonstrated how such sets can greatly speed-up MS/MS database searches. However, while the sizes of covering sets may vary between spectra, Tanner et al., 2005 [TSF$^+$05] did not describe an approach for selecting (the varying number

[1]The total number of gaps and amino acids in the gapped peptide. For example, the length of [186]DK[246]FK is 6.

of) tags for every spectrum and did not assign rigorous probabilities to tags. While Gapped Spectral Dictionaries can be utilized for generating (varying number of) conventional peptide sequence tags along with their probabilities, Table 3.1 (c) illustrates that "good" peptide sequence tags (representing all peptides in the Gapped Spectral Dictionary) may be difficult to find. We show that, contrary to conventional wisdom, some high quality spectra do not have good peptide sequence tags. We therefore advocate generating *gapped* tags representing sequences of mass gaps (like [186]LK derived from the first peptide in Table 3.1 (c)) and demonstrate that gapped tags improve the filtration efficiency of peptide sequence tags in tag-based MS/MS database searches.

Figure 3.1 illustrates different modules of MS-GappedDictionary that are described below.

3.2 Methods

3.2.1 Path Dictionary Problem.

Most *de novo* peptide sequencing algorithms interpret spectra by analyzing paths in *spectrum graphs* [DAC+99]. We start by discussing the problem of finding suboptimal paths in *arbitrary* graphs and later describe how it relates to finding paths in the spectrum graphs.

Let $G(V, E, score, probability)$ be a *directed acyclic graph* with vertex set V, edge set E, and functions *score* and *probability* defined on its edges (Figure 3.3, left panel (a)).[2] Given a path in G, the *score* of the path is defined as the sum of scores of its edges, while the *probability* of the path is defined as the product of probabilities of its edges. Given a graph G with selected vertices s (*source*) and t (*sink*), and a threshold $MinScore$, the *Path Dictionary* (denoted as $PD(G, MinScore)$) is defined as the set of all paths from s to t with scores exceeding $MinScore$ (along with their probabilities). The following Path Dictionary Problem can be solved using standard algorithms for finding suboptimal paths [Epp98].

[2] At this point, the *score* and *probability* should be viewed as arbitrary numbers assigned to the edges. Later, we will describe what *score* and *probability* mean in the context of *de novo* peptide sequencing.

`Path Dictionary Problem.` Given a directed acyclic graph G and a threshold $MinScore$, construct $PD(G, MinScore)$.

Define the *generating function* $p(x)$ as the total probability of all paths of score x from the source s to the sink t in the graph G. The generating function can be efficiently computed as the probability of node (t, x) in the *dynamic programming graph* as described in [KGBP09, KGP08] (Figure 3.3, left). $PD(G, MinLength)$ is constructed by standard *backtracking* in the dynamic programming graph.

For the *spectrum graph* of a tandem mass spectrum [DAC+99], the Path Dictionary Problem corresponds to *de novo* peptide sequencing problem when multiple (suboptimal) *de novo* reconstructions (rather than a single one) are generated.[3] Kim et al., 2008 [KGP08] applied the generating function approach (Figure 3.3, left) to analyze MS/MS spectra and further demonstrated [KGBP09] how to generate the Path Dictionary (termed *Spectral Dictionary*) that contains *all* plausible *de novo* reconstructions for a given spectrum. Each path in Path Dictionary corresponds to a full-length peptide reconstruction in the Spectral Dictionary, and $\sum_{x>MinScore} p(x)$ corresponds to the *spectral probability* (p-value) defined in [KGBP09]. To generate the Spectral Dictionaries, a spectral probability $Threshold$ is fixed and $MinScore$ is selected in such a way that the spectral probability does not exceed $Threshold$.

This Spectral Dictionary approach, while useful, is not practical for long peptides (15 amino acids and longer) with large dictionaries. We bypass this problem by solving the *Gapped Path Dictionary Problem* defined below.

3.2.2 Gapped Path Dictionary Problem.

Let H be a subset of vertices of a graph G containing the source s and the sink t (vertices of H are called *hubs*). We remark that every path on vertices in G induces a *hub path* on vertices in H by simply retaining only vertices from H in the

[3] In the spectrum graph of a spectrum, vertices represent all (integer) masses from 0 to parent mass of the spectrum, and vertices v and v' are connected by a directed edge (v, v') if and only if there is an amino acid with (integer) mass $(v' - v)$. The *score* of the edge (v, v') is given by the PRM score [TSF+05] of the peak represented by the vertex v', and the *probability* is given by the probability that the amino acid represented by the edge (v, v') appears in a random database (a database with identically and independently distributed amino acids with probability 1/20).

original path. For example, a path $s \to v_1 \to v_2 \to v_3 \to v_4 \to v_5 \to v_6 \to t$ that contains hubs s, v_2, v_3, v_5, t induces a hub path $s \to v_2 \to v_3 \to v_5 \to t$. We define the probability of a hub path as the total probability of all paths inducing this hub path. The Gapped Path Dictionary $GPD(G, H, MinScore)$ is defined as the set of all hub paths induced by the paths in $PD(G, MinScore)$ (along with their probabilities).

`Gapped Path Dictionary Problem.` Given a directed acyclic graph G, a subset of its vertices H, and a threshold $MinScore$, construct $GPD(G, H,$ $MinScore)$.

The brute-force algorithm for constructing $GPD(G, H, MinScore)$ (by constructing $PD(G, MinScore)$ and generating all hub paths induced by the paths in $PD(G, MinScore)$) is impractical for large $PD(G, MinScore)$. Below we describe an efficient algorithm for solving the Gapped Path Dictionary Problem that does not require the construction of $PD(G, MinScore)$.

Given hubs h and h', we define $Path(h, h')$ as the set of all paths in G between h and h' that *do not pass* through other hubs. Each path in $Path(h, h')$ is characterized by its score and probability. Let $\mathcal{X}(h, h')$ be the set of scores of all paths from $Path(h, h')$ and $Prob(h, h')$ be the total probability of all paths in $Path(h, h')$. If $Prob(h, h', x)$ is defined as the total probability of all paths of score x from the set $Path(h, h')$, then $Prob(h, h') = \sum_{x \in \mathcal{X}(h,h')} Prob(h, h', x)$.

We define the *hub graph* G_H as a multigraph on the set of vertices H (Figure 3.3, right). For every $x \in \mathcal{X}(h, h')$, there exists an edge between h and h' with score x and probability $Prob(h, h', x)$.[4] The *score* and the *probability* of a path in G_H is defined as the sum of scores and the product of probabilities of its edges, respectively.

As the hub paths (on vertices in H) are induced by the paths in G, $GPD(G,$ $H, MinScore)$ is the same as $PD(G_H, MinScore)$. Therefore, the Gapped Path Dictionary Problem in G is essentially the *Path Dictionary Problem* in the hub graph G_H, and we only need to compute the scores and the probabilities of the edges in G_H to solve the Gapped Path Dictionary Problem. Below, we show how to compute $Prob(h, h', x)$ for all edges of the hub graph.

Given a hub h in the graph $G(V, E, score, probability)$, we modify the score

[4]There exists $|\mathcal{X}(h, h')|$ edges between vertices h and h' in the multigraph G_H.

function by assigning score $-\infty$ to all edges originating at all hubs other than h. Denote the resulting score function (parameterized by h) as $score(h)$. The family of score functions $score(h)$ for all hubs $h \in H$ can be used to compute $Prob(h, h', x)$ for all pairs of hub vertices h and h'. One can prove that computing $Prob(h, h', x)$ (for all $x \in \mathcal{X}(h, h')$) is equivalent to computing the generating function for a graph $G(V, E, score(h), probability)$ with source h and sink h'. Note that a single computation of the generating function from h to the sink t for the graph $G(V, E, score(h), probability)$ gives us $Prob(h, h', x)$ for all $h' \in H$ and all $x \in \mathcal{X}(h, h')$.

After constructing the hub graph G_H, $GPD(G, H, MinScore)$ can be constructed by computing *generating function* for the graph G_H and generating all paths with score exceeding $MinScore$. Figure 3.3 (right) shows an example of the Path Dictionary and the Gapped Path Dictionary.

3.2.3 Gapped Spectral Dictionaries.

So far, we represented each path in the Gapped Path Dictionary as the sequence of *edges* (rather than *vertices*) the path traverses. Since the hub graph G_H is a multigraph (that may have multiple edges of various scores between the same vertices), there can be many paths (with different scores) with identical *vertex-sets* (Figure 3.3, right panel (c)). We define the *Compact Gapped Path Dictionary*, denoted by $CGPD(G, H, MinScore)$, as the set of *vertex-sets* of paths in the Gapped Path Dictionary $GPD(G, H, MinScore)$, along with their *probabilities*, where the *probability* of each vertex-set in $CGPD(G, H, MinScore)$ is defined as the total probability of the paths in $GPD(G, H, MinScore)$ with the same vertex-set.

Generation of the Compact Gapped Path Dictionary

The Compact Gapped Path Dictionary $CGPD(G, H, MinScore)$ can be generated (albeit inefficiently) from the Gapped Path Dictionary $GPD(G, H,$
$MinScore)$ by simply representing all paths with identical vertex-sets as a single vertex-set and adding up the probabilities of all such paths. However, one can efficiently generate the Compact Gapped Path Dictionary without explicitly constructing

the Gapped Path Dictionary. Since the Gapped Path Dictionary in G is the same as the Path Dictionary in the multi-graph G_H, the Compact Gapped Path Dictionary in G is the same as the vertex-sets of Path Dictionary in G_H, It is easy to see that generating these vertex-sets can be achieved by retaining only the edges with the highest scores among parallel edges in the (multi)graph G_H and constructing the Path Dictionary in the resulting (simple) graph. The Path Dictionary in this modified G_H induces the vertex-sets of the Compact Gapped Path Dictionary in G.

After the Compact Gapped Path Dictionary is generated, one still needs to compute the probability of each vertex-set. This can be done by applying MS-GeneratingFunction to a graph consisting of a single path corresponding to each vertex-set in the Compact Gapped Path Dictionary (this path represents a multi-graph since it may contain parallel edges).

Generation of the Gapped Spectral Dictionary

For each spectrum, we construct its spectrum graph and generate a set of hubs (prefix masses). Given a spectrum graph G and a set of hubs H, paths in G correspond to peptides while vertex-sets in G_H correspond to *gapped peptides* introduced in [KBP09]. *Gapped Spectral Dictionary* is defined as Compact Gapped Path Dictionary of the spectrum graph. The probability of the gapped peptide represented by a vertex-set is given by the sum of the probabilities of all edge-paths (with the same vertex-set) with scores exceeding $MinScore$.

While we described an algorithm for constructing the Gapped Spectral Dictionary for a given hub set H, it remains unclear how to select hubs. The hub selection has to achieve two conflicting goals: (i) minimize the number of selected hubs to ensure that the Gapped Spectral Dictionary is small, and (ii) maximize the average length of peptides in the Compact Gapped Spectral Dictionary to ensure that the reconstructed gapped peptides are sufficiently informative.

Therefore, the goal is to select k hubs that maximize the average number of vertices per path in the Gapped Path Dictionary (weighted by their probabilities). We select hubs as k most "popular" vertices in paths from $PD(G, MinScore)$. Such ranking of vertices of the graph G can be computed by generating *Spectral Profiles* introduced in [KBP09].

3.3 Results

3.3.1 Datasets

We used the previously published Shewanella, HEK, and Standard datasets to benchmark MS-GappedDictionary (see [GTJ+07], [TSN+07, FBS+08], and [KEH+08] for the details of the generation of spectra in Shewanella, HEK, and Standard datasets, respectively).

`Shewanella` dataset. To benchmark the performance of MS-GappedDictionary, we adopted the Shewanella dataset composed of 18,468 charge 2 spectra from *Shewanella oneidensis* MR-1, each representing a distinct tryptic digested peptide [GTJ+07].[5] The spectra in this dataset were acquired on an ion trap MS (LCQ, ThermoFinnigan, San Jose, CA) using ESI and were identified with InsPecT⊕MS-GeneratingFunction [TSF+05, KGP08] to ensure that all Peptide Spectrum Matches (PSMs) have spectral probabilities below 10^{-9}. Note that MS-GeneratingFunction was shown to improve upon other MS/MS identification tools (InsPecT, X!Tandem, and SEQUEST/PeptideProphet [KGP08]) and in most applications, peptide identifications with spectral probabilities above 10^{-9} are of little use since they result in high FDR. The analysis below is based on Shewanella dataset unless noted otherwise.

`Standard` dataset. Shewanella dataset is inadequate for benchmarking the (gapped) tag generation accuracy, since the tag-based tool InsPecT was used to identify the spectra in Shewanella dataset (i.e., a correct InsPecT tag was generated for every spectrum). We obtained the dataset reported in [KBP09] collected from the Standard Protein Mix database [KEH+08]. For this study, we considered only the charge 2 spectra generated by LTQ, where the spectra were identified by SEQUEST [EMY94] and PeptideProphet [KNKA02] that don't use tags for identifications. We further selected PSMs with spectral probabilities below 10^{-9} and formed the dataset (denoted *Standard*) with 990 charge 2 spectra of distinct peptides.

`HEK` dataset. To benchmark MS-GappedDictionary, MS-Dictionary [KGBP09], InsPecT [TSF+05], and OMSSA [GMK+04] in MS/MS searches of huge databases, we analyzed the previously published spectral dataset from the human

[5] While this paper focuses on doubly-charged spectra, the same generating function approach works for spectra with higher charges as shown in [KMB+10].

HEK293 cell line generated in Steve Briggs' laboratory (see [TSN+07, FBS+08] for a detailed description of this dataset). The spectra were acquired on an LTQ linear ion trap tandem mass spectrometer.

InsPecT and OMMSA were chosen for benchmarking since they represent some of the fastest MS/MS database search tools.[6] We selected 1 million spectra from *HEK293* dataset (described in [TSN+07]) for analyzing proteogenomics applications of MS-GappedDictionary. Since analyzing 1 million spectra even with fast tools like InsPecT is very time consuming (estimated CPU time in the search against the 6-frame translated human genome is 9 million seconds) we further selected a single run of this dataset ($\approx$ 30,000 spectra) for benchmarking. We further processed this dataset with PepNovo+ (Release 20091029) [Fra09] to correct charges and parent masses and limited our analysis to 14,000 charge 2 spectra (denoted *HEK* dataset). The HEK dataset was searched against the six-frame translation of the repeat-masked human genome (version GRCh37 released on March 2, 2009) using MS-GappedDictionary, MS-Dictionary, InsPecT, and OMSSA.

To generate the Gapped Spectral Dictionaries, the spectral probability threshold is set to 10^{-9} for Shewanella and Standard datasets and 10^{-11} for HEK dataset (assuming that the precursor mass is known). [7] The spectral hubs are selected based on k maximal peaks in its Spectral Profile with k varying from 20 to 40.

From Gapped Spectral Dictionaries to Pocket Dictionaries.

Since multiple peptides often induce the same gapped peptide, Gapped Spectral Dictionaries are typically much smaller than Spectral Dictionaries. Figure 3.4 shows the sizes of Gapped Spectral Dictionaries and Spectral Dictionaries for various peptide lengths. While the size of Spectral Dictionary grows as $20^{\text{peptide length}}$, the size of the Gapped Spectral Dictionary is limited by $2^{|H|}$, where $|H|$ is the number of hubs. In practice, the size of Gapped Spectral Dictionaries is much smaller than $2^{|H|}$ for sensible values of spectral probabilities. For example, for peptides of length 20,

[6] Sequest was shown to be 60 times slower than InsPecT [KGBP09] making it impractical for large proteogenomic searches.

[7] The spectral probability thresholds vary for different datasets to maintain roughly 1% FDR (see [GP09] for selection of the spectral probability threshold).

the size of the Spectral Dictionary exceeds 10^{17} while the size of the Gapped Spectral Dictionary is on the order of 10^4 (for $|H| = 20$).

Figure 3.5 shows the distribution of the lengths of the gapped peptides that are induced by the correct peptides (*correct gapped peptides*). The high average length of the correct gapped peptides (10 - 13) indicates that Gapped Spectral Dictionaries have the potential to speed up database searches.[8] Gapped peptides are classified into *short* (with length shorter than δ) and *long* (with length equal to or longer than δ), where δ is the minimum gapped peptide length threshold. Discarding short gapped peptides results in *δ-reduced Gapped Spectral Dictionary*.

A spectrum is *δ-identifiable* if its δ-reduced Gapped Spectral Dictionary contains at least one correct gapped peptide. Figure 3.6 shows the identifiability of spectra in the Shewanella dataset. For $\delta = 5$, the identifiability is higher than 99% for all peptide lengths. Figure 3.6 illustrates that there exists a tradeoff between the identifiability and efficiency of the database search controlled by the minimum length of the gapped peptide δ (increase in δ reduces the identifiability but improves the efficiency of the database search).

After generating the δ-reduced Gapped Spectral Dictionaries, we order all gapped peptides by their *coverages*, and analyze the rank of the first correct gapped peptides in this ranked list. The *coverage* of a gapped peptide is defined as the probability of the gapped peptide divided by the total probability of the peptides in the Spectral Dictionary. Figure 3.7 shows that the average rank of the best ranked correct gapped peptides does not exceed 100 even for long gapped peptides ($\delta = 5, 7, 9$). In fact, only 20 - 100 gapped peptides are typically sufficient to generate a correct peptide (Figure 3.8). As such, it suffices to generate a small subset of the Gapped Spectral Dictionary called *Pocket Dictionary* by choosing the k best-ranked gapped peptides in the δ-reduced Gapped Spectral Dictionary (k is typically 20 - 100). Figure 3.9 shows the identifiability of the Pocket Dictionaries compared to the identifiability in the (full-size) δ-reduced Gapped Spectral Dictionaries. Throughout the paper we generate Pocket Dictionaries of size 100 with $\delta = 5$ and 20 hubs that results in high identifiability.

[8]The fraction of short gapped peptides (length less than 5) is less than 0.01 regardless of the peptide length.

While we showed how to generate the *highest-scoring* gapped peptides, it is not immediately clear how to generate the *highest-probability* vertex-sets (gapped peptides) in the δ-reduced Gapped Path Dictionary. This difficulty stems from the fact that the y-axis in the DP graph (Figure 3.3. (a)) represents accumulated *scores* and not accumulated *probabilities*. To address this problem, we implemented a depth-first branch-and-bound backtracking traversal of the DP graph that uses accumulated scores to determine membership in the Pocket Dictionary and accumulated probabilities to select the highest-coverage peptides. To generate the top k highest probability vertex-sets, the algorithm maintains the accumulated probability for every suffix extension and combines it with node probabilities (Figure 3.3. (a)) to prune extensions whose maximum probability is lower than that of the current k-th ranked highest-probability peptide.

3.3.2 From Gapped Spectral Dictionaries to gapped tags

Once the Pocket Dictionary is generated, one still needs to match gapped peptides in the Pocket Dictionary against the protein database. The current version of MS-GappedDictionary uses *gapped tags* of length 3 (see below) instead of gapped peptides to speed-up searches in huge databases. This is conceptually similar to InsPecT search with the only difference that InsPecT uses 3-aa long peptide sequence tags while MS-GappedDictionary uses gapped tags of length 3 for filtering the database.

Table 3.1 (c) demonstrates that many gapped peptides in the Gapped Spectral Dictionary may not contain peptide sequence tags. In contrast, allowing a single gap in tags (*gapped tags*) reveals a covering set of only 6 tags of length 3: [273]LK, G[242]K, S[299]K, [250]SG, ELK, and [186]LK. In contrast with peptide sequence tags, gapped tags include both gaps and amino acid masses. Below we limit our analysis to gapped tags with gaps below 500 Da[9] and analyze gapped tags of length 3 with at most one gap (i.e., gapped tags with at least 2 amino acids). Such tags are called *proper* gapped tags. We demonstrate that the proper gapped tags have better *filtration efficiency* (defined below) than peptide sequence tags.

Some masses in a gapped peptide may represent either an amino acid or a

[9]We limit the mass of the largest gap to limit the memory requirements of MS-GappedDictionary.

gap because 5 amino acids (N, Q, K, R, and W with masses 114, 128, 128, 156, and 186, respectively) have *composite* masses equal to the (integer) sum of two amino acid masses.[10] For example, the composite mass 114 Da could represent either N or GG. Therefore, to generate a set of proper gapped tags, one has to decide whether a composite mass in the gapped tag corresponds to a single amino acid. To decide this, one can check whether a composite mass represents an amino acid by examining the hub set. A mass m is a *submass* of a composite mass $mass$ if both m and $mass - m$ represent masses of amino acid. If $mass$ starts at position $prefixMass$ in a gapped peptide, then it represents an amino acid if and only if $prefixMass + m$ represents a hub for each submass m of mass $mass$.

To generate the set of proper gapped tags, we select at most one proper gapped tag from each gapped peptide in the Pocket Dictionary. We distinguish between terminal tags (that start at N-terminus or end at C-terminus) and internal tags. The tag generation algorithm attempts to generate a proper tag for each gapped peptide from the Pocket Dictionary $P_1, \cdots, P_n$ ordered in the decreasing order of peptide coverages. At the i-th stage, the algorithm selects one proper gapped tag from peptide P_i unless (i) the peptide P_i contains one of the previously chosen proper gapped tags, or (ii) peptide P_i does not have proper gapped tags. If there are multiple proper gapped tags available for selection at the i-th stage, the algorithm selects an internal tag with the best filtration efficiency (if available), otherwise, it selects the terminal tag (with the better filtration efficiency if more than one terminal tags are available).[11]

Figure 3.10 compares the gapped tags generated by MS-GappedDictionary with peptide sequence tags generated by InsPecT (release 20090910). With 15 (on average) proper gapped tags generated by MS-GappedDictionary, the average accuracy is 94.8% while the accuracy of InsPecT tags is only 87.2% with 15 peptide sequence tags and 94.7% even with 50 tags.[12] MS-GappedDictionary constructs a table of proper gapped tags as described above. Once the table is built, finding pep-

[10] In this paper, we focus on ion-trap spectra and thus limit our analysis to integer amino acid masses. However, the generating function approach can be easily adjusted to more accurate mass measurements (see [KGP08]).

[11] Internal proper gapped tags are preferred since they typically have better filtration efficiency than terminal tags.

[12] The accuracy of tag generation is defined as the percentage of cases when the set of generated tags contains a correct tag.

tides matched to a proper gapped tag is fast, and the search space for further analysis is limited to only those matched peptides. We define the *filtration efficiency* of a peptide sequence tag/gapped tag/peptide as the ratio of the number of its matches in the random database over the database size. While the filtration efficiency of a peptide (i.e., an amino acid sequence) is $1/20^{\text{peptide length}}$ (and the filtration efficiency of amino acid is 1/20), it is easy to see that the filtration efficiency of a gap of mass m is the sum of filtration efficiencies of all amino acid sequences with mass m. It turns out that large masses typically have better filtration efficiencies than amino acids.[13] This improvement translates into a superior filtration efficiency of gapped tags as compared to peptide sequence tags (compare with [BCG07] where database searches with similar gapped tags were introduced).

For each spectrum in Standard dataset, we generated tags using MS-GappedDictionary (15 proper gapped tags per spectrum on average) and InsPecT (50 peptide sequence tags per spectrum), and measured the number of matches against the Swiss-Prot database. While InsPecT reported ≈ 2 thousand peptide sequence tag matches per spectrum on average, MS-GappedDictionary reported only ≈ 420 gapped tag matches.[14] The running time to search the Swiss-Prot database was 0.36 sec for MS-GappedDictionary (including the generation of the Gapped Spectral Dictionary and the gapped tags) and 0.51 sec for InsPecT per spectrum on a desktop machine with a 2.67-GHz Intel processor.

3.3.3 Database search with Gapped Spectral Dictionaries

To compare MS-GappedDictionary with other database search tools (for searches in huge databases), the HEK dataset was searched against the six-frame translation of the human genome (2.8 billion amino acid residues) using MS-GappedDictionary,

[13]For example, gap mass [57] (integer mass of Gly) appears in $\frac{N}{20}$ positions in a random database of size N while gap mass [400] appears in $\approx \frac{N}{121}$ positions. There are 1,102 combinations of amino acids for the gap mass [400]: 42 combinations of 3 amino acids, 664 combinations of 4 amino acids, 300 combinations of 5 amino acids, and 96 combinations of 6 amino acids. Thus, the filtration efficiency of the gap mass [400] is $42 \cdot (1/20)^3 + 664 \cdot (1/20)^4 + 300 \cdot (1/20)^5 + 96 \cdot (1/20)^6 = 0.0095$.

[14]The number of peptide matches reported by MS-GappedDictionary is only about 4 - 6 when the gapped *peptides* (not gapped tags) in the Pocket Dictionaries (with size 100) are used for the same experiment. The filtration efficiency of a gapped peptide, therefore, is 10^6 - 10^7 times better than that of gapped tags or peptide sequence tags.

MS-Dictionary (ver. 20100415) [KGBP09], InsPecT (release 20090910) [TSF+05], and OMSSA (ver. 2.1.7) [GMK+04]. We plotted the peptide level FDR curve of each tool in this search using the target-decoy database approach as described in [EG07]. In the case of MS-GappedDictionary, two different methods to search in the database are used: the search with gapped *tags* and the search with gapped *peptides*. We use a brute-force scanning algorithm for matching gapped peptides against the database.[15].

To measure the FDR of each tool, we first generated the reversed decoy database of the six-frame translation of the human genome. The spectra in HEK dataset were searched against both the target and decoy databases. Figure 3.11 shows the FDR curve of each tool and illustrates that MS-GappedDictionary significantly improves on all other tools in the number of reliably identified peptides for all levels of FDR ($\approx$ 30% improvement in the case of 1% FDR). InsPecT is shown to improve on OMSSA and MS-Dictionary. However, MS-GappedDictionary is $\approx$ 20 times faster than InsPecT (0.8 sec vs 17 sec per spectrum, respectively) [16] [17]. OMSSA and MS-Dictionary are also fast (1.2 sec and 0.8 sec per spectrum, respectively) but their FDRs deteriorate significantly in comparison with MS-GappedDictionary.

Figure 3.12 shows the length distribution of peptide identifications in the HEK dataset identified with MS-Dictionary and MS-GappedDictionary (in searches against the six-frame translation of the human genome). While Both tools identified roughly the same number of *short* peptides (length less than 14 aa), MS-GappedDictionary significantly improves on MS-Dictionary in identifying *long* peptides (14 aa and longer). This is a consequence of the fact that MS-Dictionary has to truncate the (large) spectral dictionaries of long peptides resulting in loosing many peptide identifications.

In contrast to MS-Dictionary, peptides matched to gapped peptides or gapped

[15]Searching gapped peptides against a database can be done by simply scanning each gapped peptide in the Pocket Dictionary against the database. Since a more efficient search with gapped peptides will be described elsewhere, the goal of this search with gapped peptides is to study FDR rather than to establish the running time of this primitive approach.

[16]All tools used in this benchmarking preprocess the protein database. Since preprocessing time is negligible (compared to the search time), we do not report the database preprocessing times. The running times include both target and decoy database search times. Except OMSSA, the six-frame translation of the human genome should be divided into small sub-databases due to the memory overhead (in MS-Dictionary and MS-GappedDictionary) or unexpected errors (in InsPecT). The running time of each tool is measured by summing the search times on the sub-databases.

[17]MS-GappedDictionary filters out poor quality spectra [FBS+08] and does not generate their Gapped Spectral Dictionaries.

tags generated by MS-GappedDictionary may not belong to the Spectral Dictionary. For example, a gapped peptide AT[144]GG may match to ATSGGG (in the Spectral Dictionary) and ATGSGG (not in the Spectral Dictionary). Thus, all peptides matched by MS-GappedDictionary have to be scored to remove those that are not in the Spectral Dictionary.[18] Since the number of peptides matched by MS-GappedDictionary before scoring is typically small, the time required for removing low-scoring peptides is negligible (less than 0.01 s per spectrum).

3.3.4 Proteogenomics application

We searched 1 million spectra from the *HEK293* dataset against the six-frame translation of the human genome (version GRCh37 released on March 2, 2009). The Pocket Dictionary size was set to 100 and the stringent spectral probability (p-value) for MS-GappedDictionary was set to $1.3 \cdot 10^{-13}$ resulting in the identification of 6,036 peptides with the corresponding peptide-level FDR of 1%. While such stringent spectral probability threshold significantly reduces the number of identified peptides, it ensures that we accept only high-quality peptide identifications.

5,958 out of 6,036 identified peptides (nearly 99%) also match the human IPI database demonstrating that MS-GappedDictionary can reliably identify many human peptides *without* knowing the human proteome. The fact that 99% of peptide identifications are found in the human protein database implies that MS-GappedDictionary represents a valuable tool for proteogenomics annotations that can be immediately incorporated into the Augustus gene prediction pipeline [CPS$^+$08, SKG$^+$06]. The remaining 78 peptides that do not match[19] the human IPI database (version 3.70) [KDW$^+$04] represent either erroneous identifications or new proteogenomics clues (previously unannotated coding regions). Further analysis of these peptides is not the focus of this paper and will be described elsewhere.

[18] There may be multiple peptides in the database matched to the gapped peptides or gapped tags. However, MS-GappedDictionary never accept a PSM (Peptide-Spectrum Match) without scoring the *entire spectrum* against the full length peptide using MS-GF scoring function. This additional scoring step applies to all found PSMs (gapped peptides in the Pocket Dictionary are only used to filter the database). After MS-GF scoring, MS-GappedDictionary assigns p-values (spectral probability) to each PSM.

[19] The amino acids Q/K and I/L are considered equivalent in this analysis.

3.4 Conclusion

Gapped peptides occupy a niche between accurate but short peptide sequence tags and long but inaccurate full-length peptide reconstructions. The gapped peptides are both long and accurate making them an ideal choice for *de novo*-based MS/MS database searches. In difference from peptide sequence tags, they typically have a few matches in a database often reducing peptide identification to a single look-up in the database. While future work will focus on efficient matching of gapped peptides against large databases, we show how gapped tags can be generated from gapped peptides to effectively filter indexed databases. Furthermore, we show how the concept of *coverage* can be instrumental for ranking sparse representations of spectral dictionaries, here limited to gapped tags and gapped peptides but conceptually generalizable to any sparse representation of all plausible peptide reconstructions. We emphasize that every gapped peptide search must be complemented by rigorous scoring of *all* found peptide-spectrum matches (i.e., with MS-GeneratingFunction [KGP08] as described above) to ensure that only statistically significant PSMs are reported. MS-GappedDictionary enables proteogenomics (e.g., searches against the six-frame translation of large genomes) and metagenomics (e.g., searches against 1000+ already sequenced bacterial genomes) analysis that is prohibitively slow for traditional MS/MS database search tools.

While this paper focuses on non-modified gapped peptides (proteogenomics studies are typically based on non-modified peptides, [20] MS-GappedDictionary is applicable to spectra of modified peptides as well. If the set of modifications is given in advance (like in traditional MS/MS search approaches), one can generate the set of modified gapped peptides by simply extending the set of masses to accommodate masses of modified amino acids. Nevertheless, the probability that the Pocket Dictionary contains a correct gapped peptide may start decreasing if diverse modifications are added to the analysis. Moreover, gapped peptides with modifications should be converted into those without modification when they are used for the database search.

[20] We remark that many modified peptides identified in typical MS/MS searches are also identified as non-modified peptides. For example, while oxidation of Met is very common, as observed in Gupta et al. 2007 [GTJ+07], for a great majority of identified peptides with Met^{+16}, there exists also a non-modified version of the same peptide (that is sufficient for proteogenomics applications). This observation applies to most chemical adducts and even some biological modifications.

The algorithms that address these issues are under development.

While MS-GappedDictionary has a potential to speed-up database searches by orders of magnitudes as compared to other widely used tools such as SEQUEST and InsPecT, its performance deteriorates in the case of highly charged spectra (charge 4 and higher). This is a bottleneck for all MS/MS database search approaches based on full length peptides or peptide sequence tags [TSF+05]. Further advances in design of scoring functions for highly charged spectra are needed to address this bottleneck [KMB+10].

We emphasize that the benefits of a pre-processed database are best utilized when the database does not need to be re-processed to reflect changes in enzyme specificity, number of missed cleavages, etc. Our approach assumes a standard combinatorial pattern matching (CPM) database pre-processing (e.g., hash tables, keyword trees, suffix trees, etc. [Gus97]) rather than a specialized MS/MS database pre-processing that may account for different search parameters such as the precursor mass or the enzyme specificity. Thus, we assume that applications of MS-GappedDictionary do not require database re-processing when the search parameters change. While traditional MS/MS database pre-processing (e.g., by parent mass) may be more specific than a CPM pre-processing, this benefit is being offset by the universal nature of CPM pre-processing and by the fact that gapped peptide searches are much faster than the traditional database searchs (even with universal rather than specialized database indexing). In the case when the search changes to include an additional post-translational modification, we suggest to change the gapped peptide generation (i.e., to transform gapped peptides with modifications into gapped peptides without modifications) rather than to re-process the database.

Acknowledgements

Chapter 3 was published as "Gapped Spectral Dictionaries and Their Applications for Database Searches of Tandem Mass Spectra". K. Jeong, S. Kim, N. Bandeira, and P. A. Pevzner. *Molecular&Cellular Proteomics*, vol. 10, M110.002220, 03 2011. The dissertation author is the primary author of this paper.

Table 3.1: (a) The Gapped Spectral Dictionary for the spectrum of peptide LNRVSQGK (consisting of 7 gapped peptides) is much smaller than the Spectral Dictionary (consisting of 92 full-length peptides). For simplicity, LNRVSQGK is represented by its *integer* amino acid masses as follows: [113][114][156][99][87][128][57][128]. Each gapped peptide is represented by amino acids and *mass gaps* that represent combinations of amino acids (for example, [128] can be Q, K, GA, or AG). Either Q or K is used instead of [128] when [128] occupies the same position as Q or K on the peptide LNRVSQGK. The gapped peptides that match the correct peptide are called *correct* gapped peptides (like gapped peptides 1 and 6 marked with †). For example, the gapped peptides [113 + 114]RVSQGK or LN[156+99]SQGK match peptide LNRVSQGK. The second column represents the coverage of the gapped peptide (see Results section for the definition of coverage), reflecting the portion of the total probability of all full-length peptides represented by the gapped peptide.
(b) Peptide sequence tags of length 3 derived from the Gapped Spectral Dictionary. Masses over left (right) arrows are the prefix (suffix) masses of the tags. The third column shows the coverage of each tag, where the coverage of a tag is defined by the summation of the coverages of gapped peptides covered by the tag. The fourth column shows the gapped peptides (specified by the numbers in the first column of (a)) covered by each tag. For example, a tag VRV covers two gapped peptides 3 and 5 in (a) with coverages of 13.71% and 5.71%, respectively. The coverage of the tag VRV is, thus, $13.71 + 5.71 \approx 19.4\%$. Overall, only 2 tags (e.g., QGK and VRV) cover all gapped peptides in the Gapped Spectral Dictionary.
(c) The Gapped Spectral Dictionary for the spectrum of peptide AIIDAIVSGELK shown in Figure 3.2 (b) (16 gapped peptides represent 24,034 full length peptides). The correct gapped peptides are marked by †. The Gapped Spectral Dictionary for the peptide AIIDAIVSGELK reveals only 3 tags (GEL, ELK, and SGE), together covering only 18.59% of the Spectral Dictionary. In contrast, 6 (*gapped tags*) [273]LK, G[242]K, S[299]K, [250]SG, ELK, and [186]LK cover the entire Spectral Dictionary.

No.	Gapped Peptide (GP)	Coverage of GP (%)*	# of peptides represented by GP
1†	[227]RVSQGK	45.69	12
2	[128] [255]VSQGK	15.99	32
3	[128]VRVSQGK	13.71	20
4	[128]VR[186]QGK	11.42	4
5	[128]VRV[215]GK	5.71	2
6†	[383]VSQGK	5.71	2
7	[128]G[198]VSQGK	1.77	20
Total	·	100	92

(a)

No.	Tag	Coverage of tag(%)	Covered GP
1	$\overleftarrow{569}$ QGK $\overrightarrow{0}$	94.3	1,2,3,4,6,7
2	$\overleftarrow{383}$ VSQ $\overrightarrow{185}$	82.9	1,2,3,6,7
3	$\overleftarrow{482}$ SQG $\overrightarrow{128}$	82.9	1,2,3,6,7
4	$\overleftarrow{227}$ RVS $\overrightarrow{313}$	59.4	1, 3
5	$\overleftarrow{128}$ VRV $\overrightarrow{400}$	19.4	3,5

(b)

Table 3.1, continued

No.	Gapped Peptide (GP)	Coverage of GP (%)*	# of peptides represented by GP
1	[445][250]S[186]LK	33.81	3286
2†	[695]S[186]LK	19.18	1703
3	[445][337][186]LK	13.28	255
4	[445][250][273]LK	7.67	178
5†	[782]GELK	6.10	684
6†	[695]SGELK	5.55	5563
7	[445][250]S[299]K	4.20	901
8	[445][250]SGELK	3.78	3437
9	[445][337]GELK	1.98	1072
10	[445][250]SG[242]K	1.61	3942
11†	[695]SG[242]K	0.91	1614
12	[445][394]ELK	0.91	507
13	[445][250]SG[370]	0.66	604
14	[445][250][144]ELK	0.20	91
15†	[695][144]ELK	0.07	35
16	[445][337]G[242]K	0.09	162
Total	·	100	24034

(c)

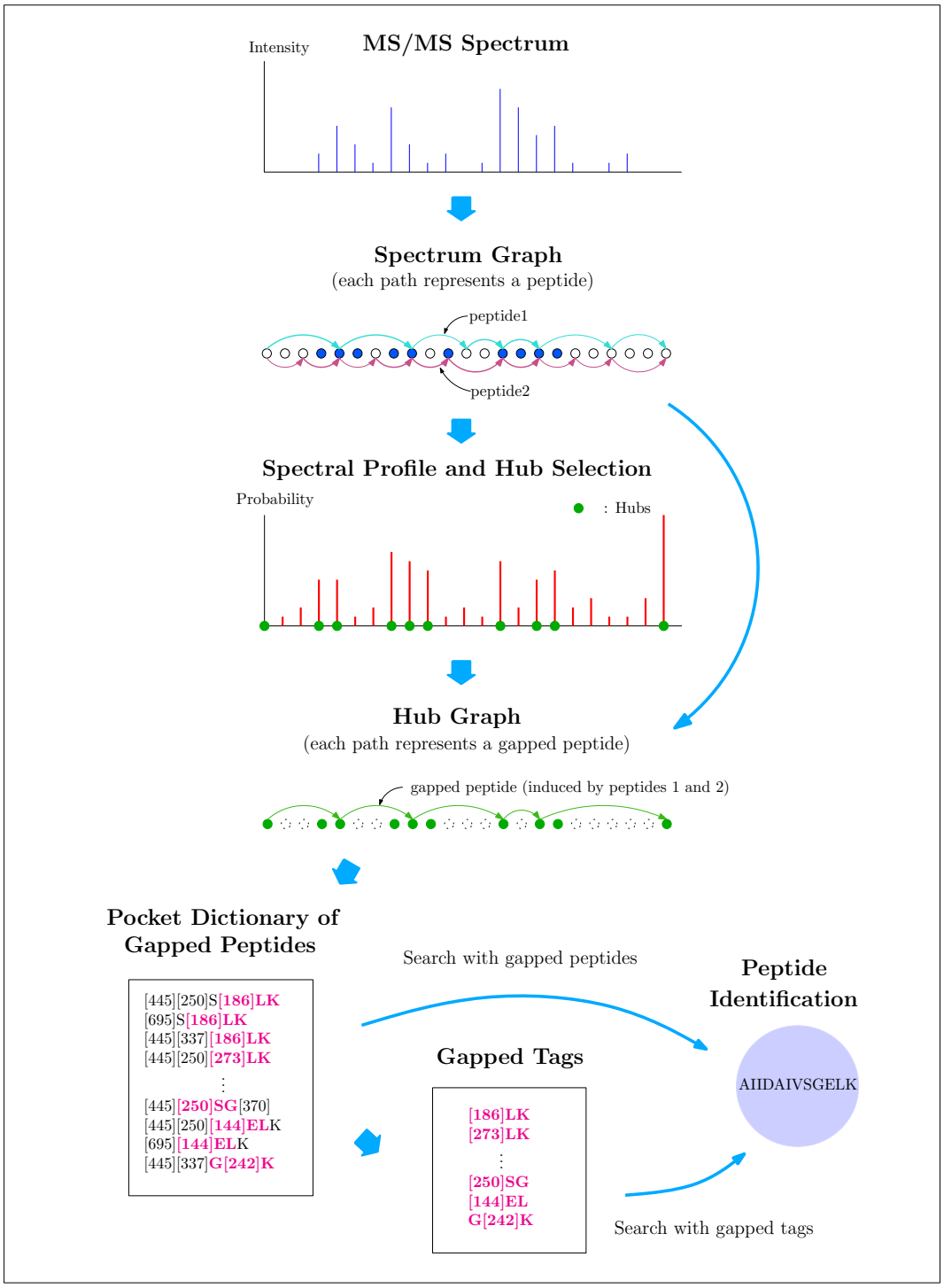
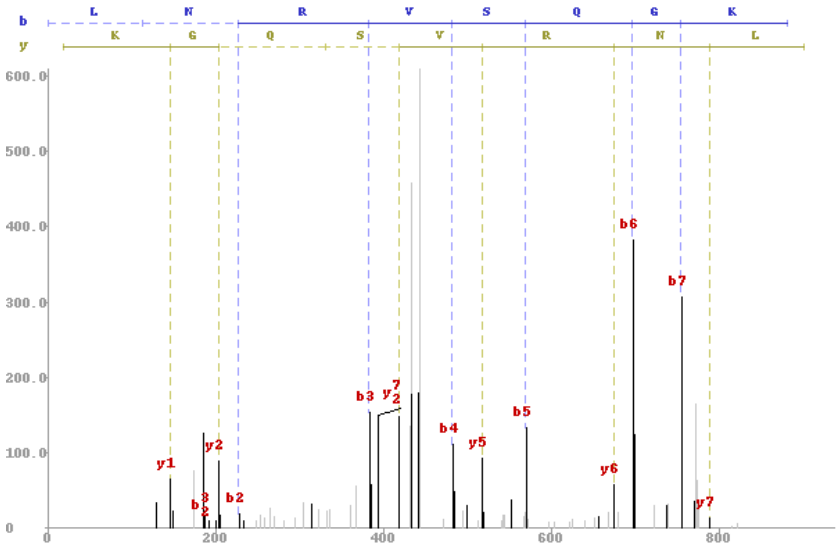
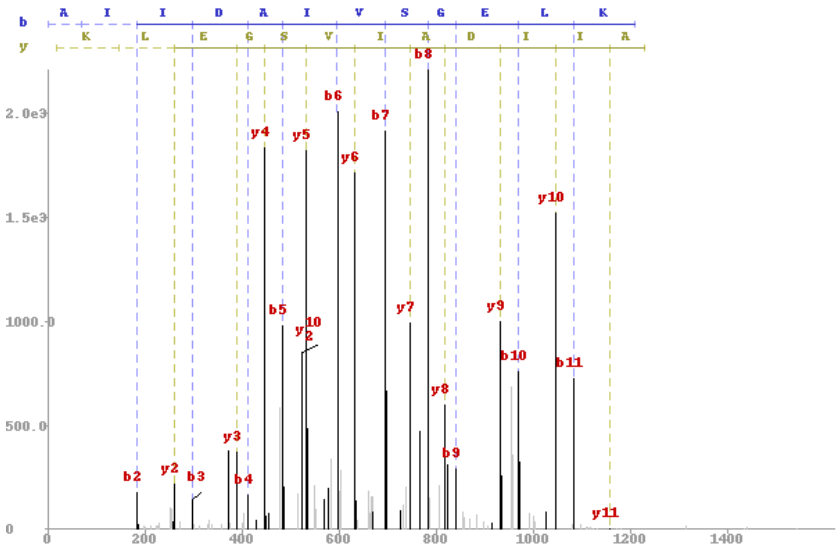


Figure 3.1: Different modules of MS-GappedDictionary.



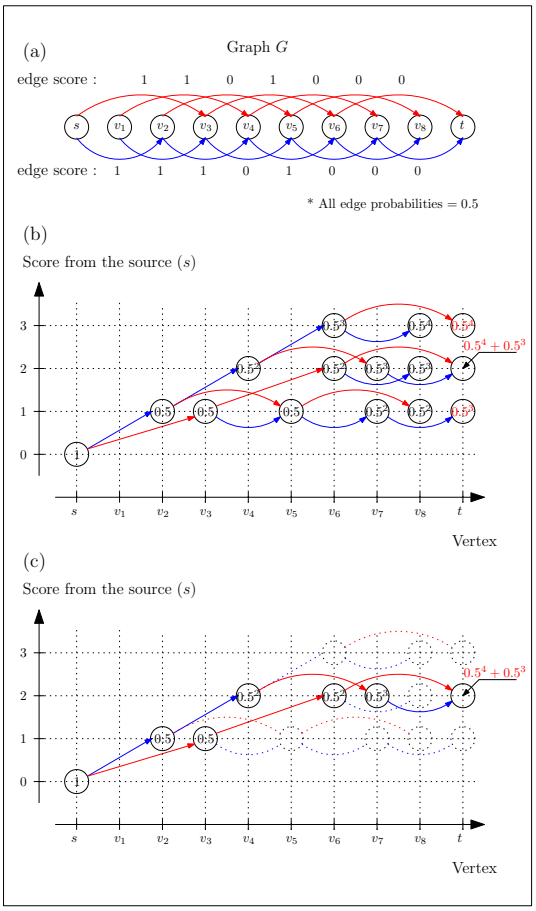
(a)

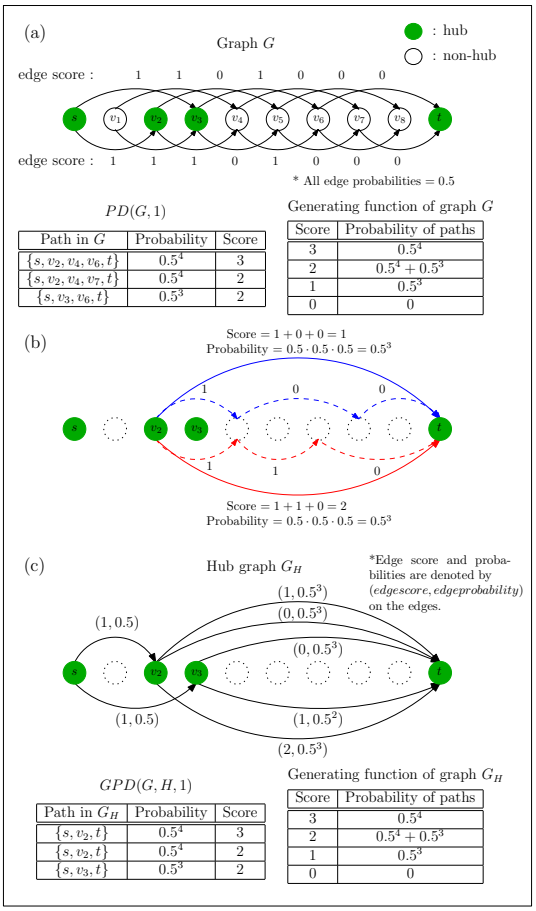


(b)

Figure 3.2: Spectra for the peptide LNRVSQGK (a) and AIIDAIVSGELK (b) identified by InsPecT (release 20090910) database search.

Figure 3.3: **Left panel :** Illustration of the dynamic programming algorithm for computing the generating function of graph G shown in (a). The nodes of the *dynamic programming (DP) graph* (b) are defined as pairs (v, x), where v is a vertex of G and x is a score. Two nodes (v, x) and (v', x') are connected by an edge if and only if there exists an edge between vertices v and v' in G with score $x' - x$. The probability of an edge between (v, x) and (v', x') in the DP graph equals to the probability of the edge (v, v') in G. A source s in graph G corresponds to a single node $(s, 0)$ in the DP graph. A node (v, x) is present in the DP graph if and only if there exist a path from $(s, 0)$ to (v, x). In this example, red (blue) edges of the DP graph in (b) are from the red (blue) edges of the graph G in (a). All edge probabilities in (b) are 0.5 as the probabilities of edges of G are 0.5. The *node probability* of node (v, x) (shown inside nodes in (b) and (c)) is the total probability of the paths from the source s to v with the score x. The node probability of the source of the DP graph is initialized by 1, and the node probability of a node (v, x) is obtained by the *weighted* summation of the node probabilities of its *predecessors* (see [KGP08]). The generating function is represented by the probabilities of the sink nodes in the DP graph. To find all paths of score x from the source to the sink in graph G one has to backtrack all paths from the node (t, x) in the DP graph. For example, if $x = 2$, two such paths are found: $\{s, v_2, v_4, v_7, t\}$ and $\{s, v_3, v_6, t\}$ as in (c).
Right panel : Path Dictionary and Gapped Path Dictionary. (a) $PD(G, 1)$ and the generating function of G. (b) The construction of G_H using edges between hubs v_2 and t (shown as solid blue and red edges) as examples. Solid blue and red edges in G_H are induced by dashed blue and red paths in G. All paths that use only non-hub vertices in G are collapsed into edges in G_H. (c) The hub graph G_H, $GPD(G, H, 1)$, and the generating function of G_H.





Path in G	Probability	Score
$\{s, v_2, v_4, v_6, t\}$	0.5^4	3
$\{s, v_2, v_4, v_7, t\}$	0.5^4	2
$\{s, v_3, v_6, t\}$	0.5^3	2

Score	Probability of paths
3	0.5^4
2	$0.5^4 + 0.5^3$
1	0.5^3
0	0

Path in G_H	Probability	Score
$\{s, v_2, t\}$	0.5^4	3
$\{s, v_2, t\}$	0.5^4	2
$\{s, v_3, t\}$	0.5^3	2

Score	Probability of paths
3	0.5^4
2	$0.5^4 + 0.5^3$
1	0.5^3
0	0

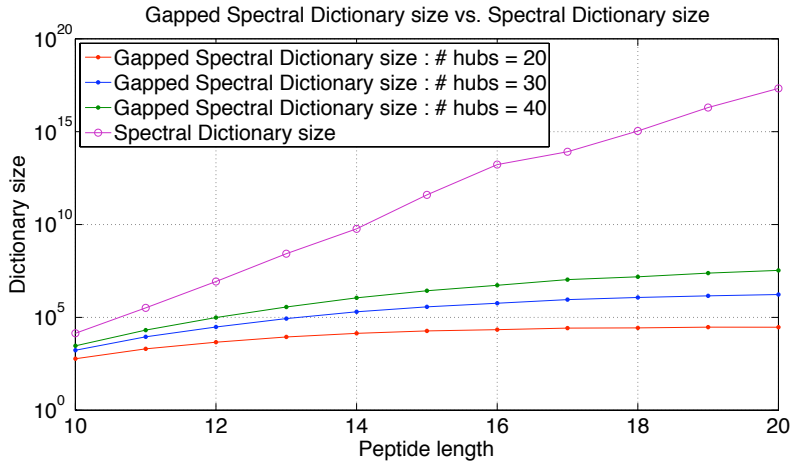


Figure 3.4: Gapped Spectral Dictionary size vs. Spectral Dictionary size (for varying peptide length and number of hubs) for the Shewanella dataset.

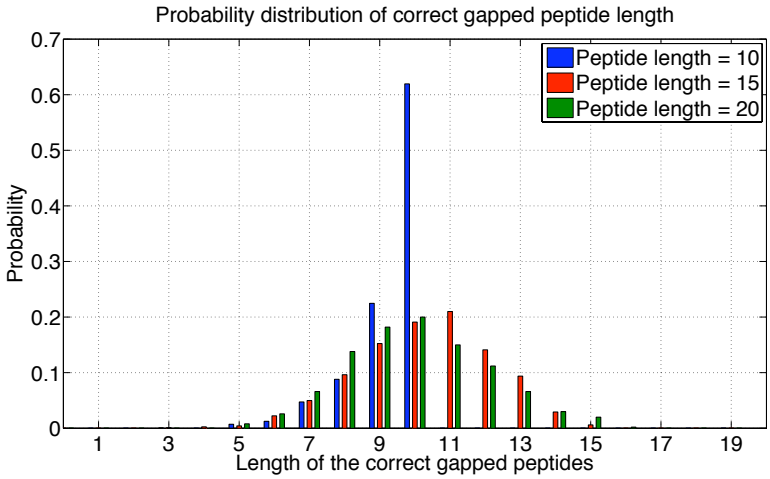
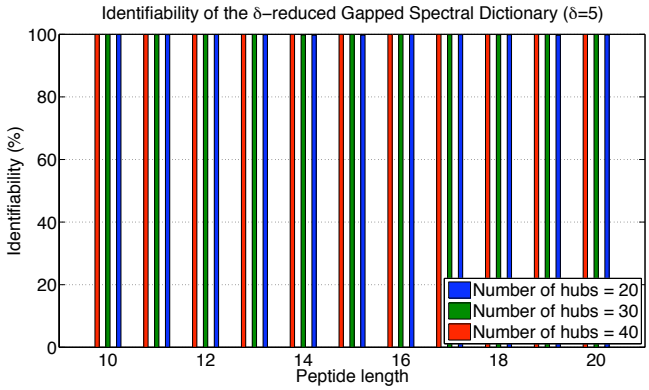
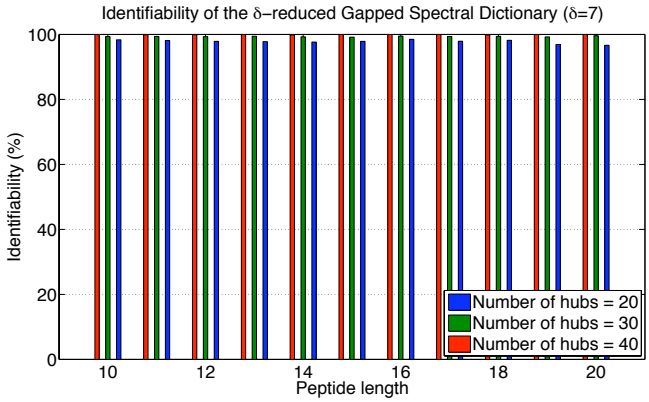


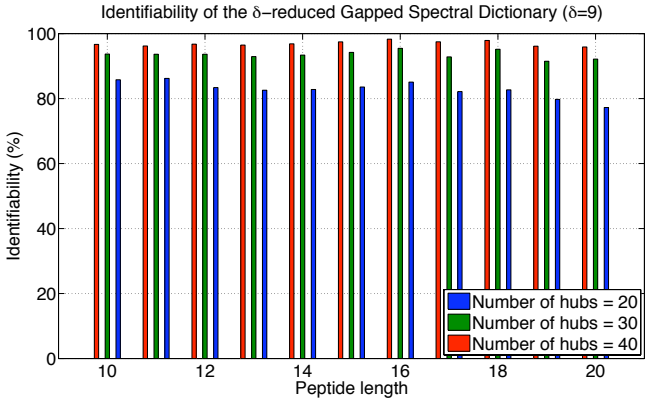
Figure 3.5: Distribution of the lengths of the gapped peptides induced by correct peptides (for 20 hubs) for the Shewanella dataset.



(a)



(b)



(c)

Figure 3.6: Identifiability of the δ-reduced Gapped Spectral Dictionaries from the Shewanella dataset for $\delta = 5$ (a), $\delta = 7$ (b), and $\delta = 9$ (c).

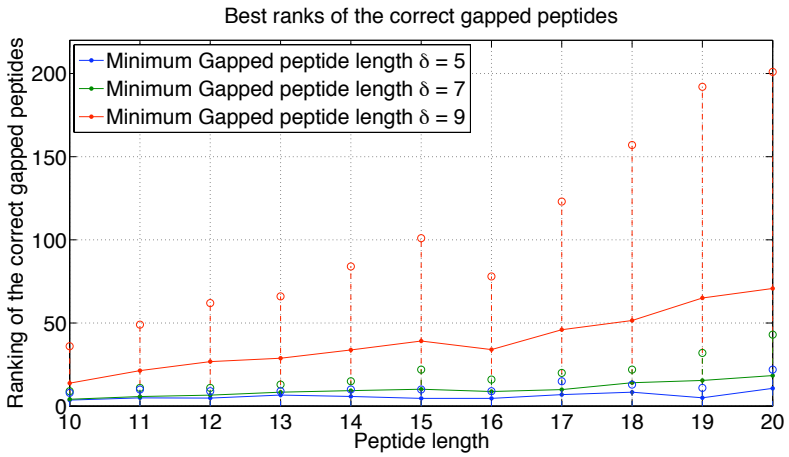


Figure 3.7: Average rank of (the best ranked) correct gapped peptides. The average ranking does not exceed 80 regardless of the peptide length (for $\delta = 5, 7, 9$). The number of hubs is 20. The dotted lines with open circles at the ends represent the range that the rankings fall into 90% of the time.

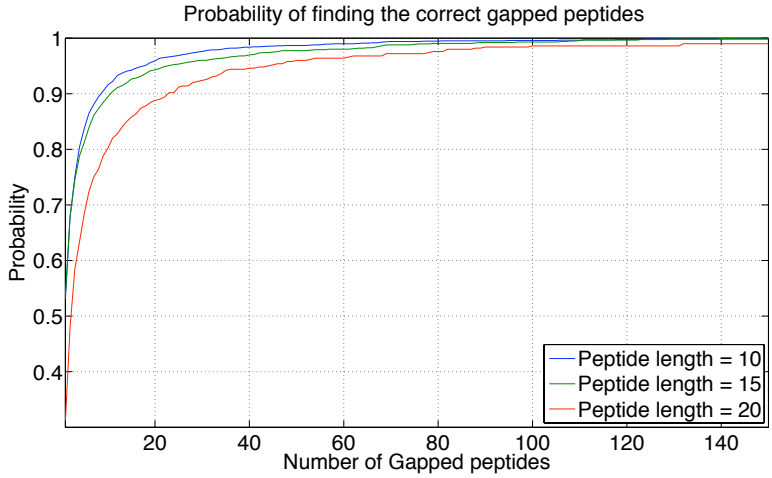
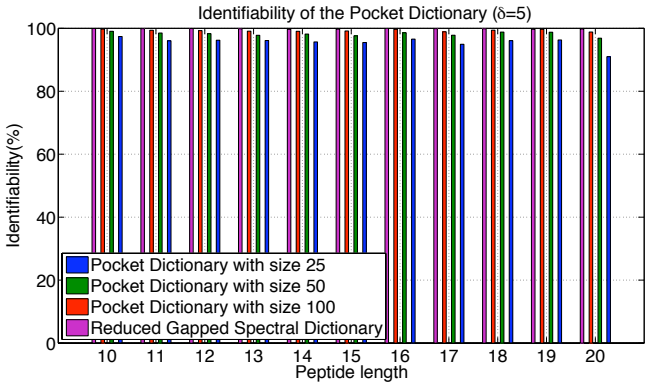
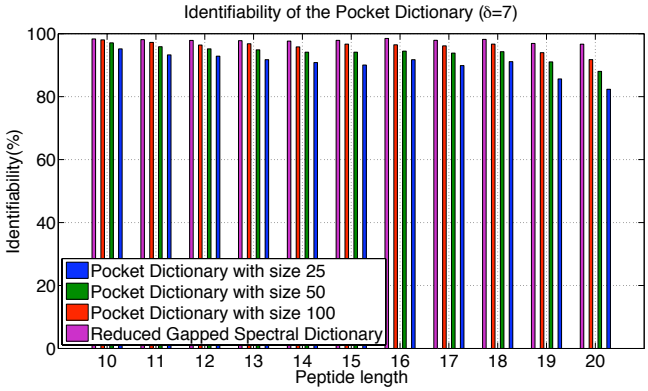


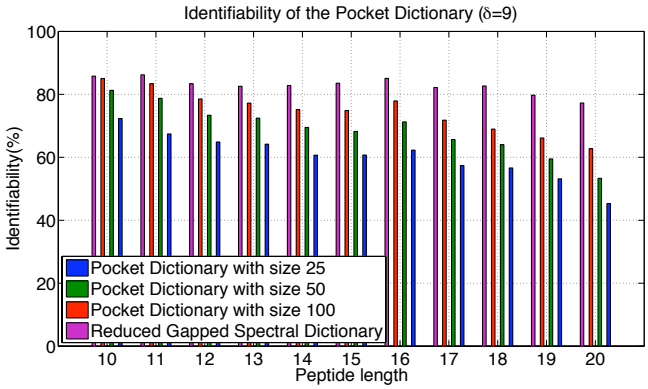
Figure 3.8: The probability that a correct gapped peptide is found within k top-ranked peptides in the δ-reduced Gapped Spectral Dictionary. The number of hubs is 20, and $\delta = 5$.



(a)



(b)



(c)

Figure 3.9: Identifiability of the Pocket Dictionaries from the Shewanella dataset for $\delta = 5$ (a), $\delta = 7$ (b), and $\delta = 9$ (c). The number of hubs is 20. Even for long peptides, Pocket Dictionaries with 50 gapped peptides are sufficient to ensure the identifiability higher than 97% when δ is 5. When δ is large, larger Pocket Dictionaries are needed.

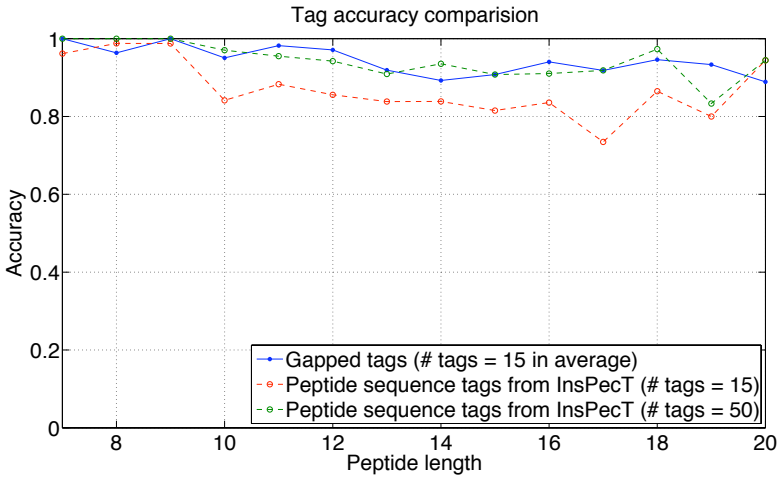


Figure 3.10: Comparison of gapped tags generated from the Pocket Dictionaries and the peptide sequence tags generated by InsPecT (on spectra from the Standard dataset).

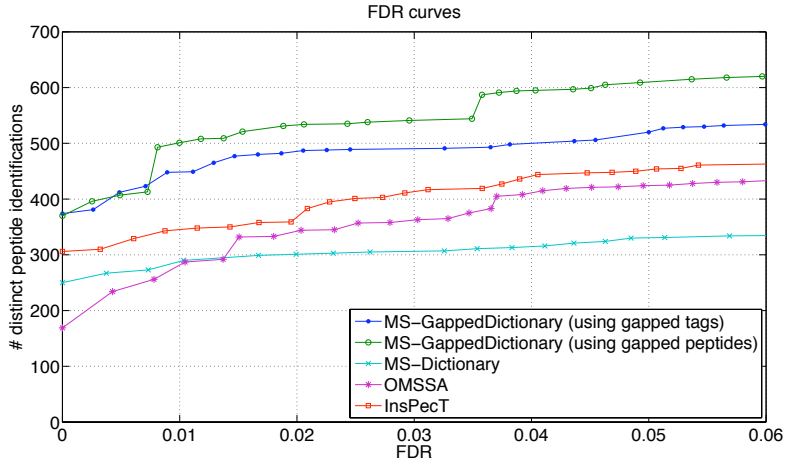


Figure 3.11: The FDR curves for MS-GappedDictionary (using either gapped tag or gapped peptides), OMSSA, InsPecT, and MS-Dictionary (peptide-level FDR is reported [EG07]). For each spectrum, only the single best matching peptide is reported.

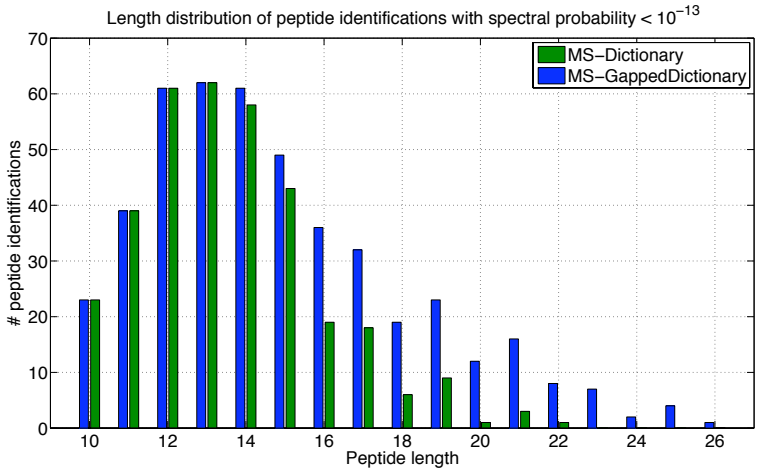


Figure 3.12: The length distribution of peptides with the spectral probability less than 10^{-13} (corresponding FDR ≈1%) in HEK dataset identified by MS-GappedDictionary and MS-Dictionary in the six-frame translation of the human genome. MS-Dictionary identifies less peptides than MS-GappedDictionary when the peptide length is longer than 13.

Chapter 4

False discovery rates in spectral identification

4.1 Introduction

Mass spectrometry (MS) based proteomics studies often generate millions of tandem mass spectra. These spectra are usually assumed to come from peptides and typically interpreted using a database search engine. There are numerous database search engines available such as SEQUEST [EMY94], Mascot [PPCC99], X!Tandem [CB04], OMSSA [GMK+04], InsPecT [TSF+05] and MS-GFDB [KMB+10]. These engines take a set of spectra and a protein database as the input and output peptide-spectrum matches (PSMs) by scoring each spectrum against the peptides in the database and assigning the best-scoring peptide as a "match" to each spectrum. In most experiments, only a small portion of these PSMs (20% - 40%) represent plausible matches [Nes10, KMB+10, TSF+05]. Therefore, identifying correct PSMs among a mixture of correct and incorrect PSMs is an important problem in MS based proteomics. Since confidence in PSM assignments is usually represented as a score, this problem is equivalent to setting up a score threshold where PSMs with scores above the threshold are regarded as *positive discoveries* (or positive PSMs) while the remaining are regarded as *negative discoveries* (or negative PSMs). The score threshold must be appropriately determined because low thresholds lead to excessive false positives and high thresholds lead to too many false negatives.

The target-decoy approach (TDA) [EG07, Nes10] is currently the most widely used strategy to address this problem. Given a protein database (target database), this approach requires that spectra be searched not just against the target database but also against a *decoy* database. A decoy database is a reversed, shuffled (e.g. permuted) or otherwise randomized database of the same size as the target database. It is assumed that the positive PSMs from the decoy database (decoy PSMs) are false and that the expected number of decoy PSMs equals the expected number of false positive PSMs from the target database. Thus, by counting the number of decoy PSMs, one can estimate the *False Discovery Rate* (*FDR*) - the proportion of false PSMs among positive PSMs. Estimating FDRs via TDA is currently the standard in high-throughput MS studies because it is simple, easily implementable, and widely applicable to various experimental set-ups while successfully distinguishing correct and incorrect PSMs.

However, there is no consensus on the exact procedure for TDA - a worrisome situation since the quality of the resulting FDR estimates and the number of resulting PSMs are strongly dependent on such procedural variations. For example, there are multiple methods to generate decoy databases (e.g., it could be a reversed, shuffled or randomized version of the target database) but it remains an open problem to determine the optimal way of generating and using decoy databases. Also, it is questionable whether to search the target and the decoy database separately or to search the concatenated target and decoy databases. Furthermore, even after the score threshold is determined, it is ambiguous what formula to use to calculate the FDR. Because of all these "variations", the same FDR (e.g. FDR 1%) may mean a different confidence level depending on the specific procedure, and it is often difficult to determine how much trust can be allowed for FDRs reported in research papers on MS studies.

We compare various TDA procedures and assess them in terms of how accurate they estimate the "true" FDRs and how many PSMs they identify at a fixed true FDR. We also show how different database search parameters such as the the choice of the protein database, parent mass tolerance and enabling/disabling two-pass searches affect the accuracy of FDR estimation and the resulting set of PSMs. Based on our results, we recommend a set of TDA guidelines and search parameters towards

improving the accuracy of FDR estimates while also producing more resulting PSMs.

We used X!Tandem [CB04] and MS-GFDB [KMB+10] as the database search engines. The conclusions presented here should apply to most other database search engines but may vary depending on particular implementation and design details.

4.2 Materials

4.2.1 MS/MS Spectra

The main MS/MS spectra dataset used in this study was the LTQ-Orbitrap dataset in Mix 7 from the *ISB Standard Protein Mix Database* [KEH+08]. It consists of 47,292 spectra (denoted by **ISB-All**) from 10 replicates generated from tryptic digests of 18 proteins called *ISB Standard Protein Mix*. For most experiments, a subset containing 4,966 spectra from replicate 02 (denoted by **ISB-02**) were used.

We also analyzed the *Study 6 LTQ-XL-Orbitrap@86* data set generated by the clinical proteomic technology assessment for cancer (CPTAC) network [PBH+]. This dataset consists of LTQ-Orbitrap spectra from tryptic digests of yeast proteins with Sigma UPS1 spiked in. From the original dataset, we took 124,193 spectra to form **Y-All** dataset and further randomly selected 9,758 spectra out of **Y-All** dataset to form **Y-Small** dataset.

To compute factual FDRs (to be defined below), we additionally obtained a dataset of monoclonal antibody spectra from a previous protein sequencing study by Bandeira et al. [BOMP08] consisting of 19,982 spectra (denoted by **AB-All**). Among them, 6,319 Spectra from trypsin and chymotrypsin digests (denoted by **AB-TC**) were mainly used for most experiments.

4.2.2 Protein Database

We used the protein database of ISB Standard Protein Mix (18 proteins, 7,440 amino acids, denoted by **ISB**) for the ISB-All and ISB-02 datasets and the yeast database (from Ensembl ftp://ftp.ensembl.org, release 60, 6696 proteins, 3,011,992 amino acids, denoted by **Yeast**) for the Y-All and Y-Small datasets.

We also obtained an Arabidopsis thaliana database from the Arabidopsis Information Resource (TAIR) (http://arabidopsis.org, release 9, 33,410 proteins, 13,468,323 amino acids). The Arabidopsis thaliana database (denoted by **AT**) was also used to compute factual FDRs.

4.2.3 Database Search Engine

We used X!Tandem (version 12/01/2011) [CB04] and MS-GFDB (version 01/06/2012) [KMB+10] as database search engines. For both engines, the parent mass tolerance was set to either 2.5 Da or 30 ppm (parts per million) according to the experiment (see Table 4.1). When the parent mass tolerance was 30 ppm, we allowed isotopic mass errors (i.e., +1, +2 and -1 Da errors) in the parent mass because such errors are very common for LTQ-Orbitrap spectra. We used the spectral probability for MS-GFDB and the hyper score for X!Tandem to score PSMs unless otherwise noted. Only the best match per spectrum was reported and no spectrum quality filter was used. For X!Tandem, the fragmentation ion tolerance was set to 0.5 Da and the two-pass search was deactivated.

4.3 Methods

The most commonly used TDA procedure (denoted by **Standard TDA Protocol**) is as follows:

Given a set of spectra, a protein database (target database) and a database search engine,

1. Generate a decoy database by reversing the target database.
2. Concatenate the target and decoy database and run a database search engine against the concatenated database. For each spectrum, consider only the best scoring (either target or decoy) PSM.
3. Sort all PSMs in decreasing (or increasing) order of match scores (or E-values/p-values).

4. For a threshold t, estimate the FDR as N_{decoy}/N_{target} where N_{target} (N_{decoy}) is the number of positive target (decoy) PSMs (i.e., PSMs with scores better than t).

5. Report the set of target PSMs with scores better than t and a corresponding FDR.

Although the above TDA procedure is frequently used, many researchers do not follow exactly these steps. For example, instead of using the reversed database, some generate decoy databases by shuffling protein sequences in the target database citetanner05 or enumerating amino acids randomly. Also, some prefer to run database searches separately for the target and decoy database and consider two PSMs per spectrum (one from the target database and the other from the decoy database). Moreover, some use $2 \cdot N_{decoy}/(N_{target} + N_{decoy})$ instead of N_{decoy}/N_{target} as the formula to compute FDRs. Depending on such choices, the set of resulting PSMs is very likely to vary.

In addition to the specific TDA procedure, one may get significantly different resulting PSMs depending on the choice of the protein database and search parameters. Below, we evaluate how these factors affect FDR estimation and change the resulting set of PSMs. In particular, we address the following cases/issues:

1. How to construct a decoy database: reversed vs shuffled

2. Concatenated vs separate decoy

3. Choice of formula to calculate FDR

4. Impact of the size of the database

5. How the number of spectra affects the results

6. Expected gains from accurate peptide parent masses

7. How the score normalization affects the results

8. PSM-level vs Peptide-level FDR

9. Two-pass searches and TDA

To address these issues, we designed a set of experiments by varying the set of spectra, protein database, TDA procedure, and search parameters (Table 4.1). For each experiment we measure how accurate the FDR estimation is by measuring the *factual FDR*. The *factual FDR* is defined as follows: If we are given a dataset where all spectrum identifications are perfectly known (a *fully-labeled approach*) in advance then one can easily validate the FDR estimated via TDA (denoted by *empirical* FDR) because it would be possible to compute the "true" FDR. But since such a dataset is not readily available, similar to Granholm et al. [GNK], we use a *semi-labeled approach* where false PSMs (termed *dummy* PSMs) are intentionally introduced using the following three ways.

1. Dummy databases: Let *dummy proteins* be the proteins from which the searched spectra are not supposed to be generated. The *dummy database* is a database containing only dummy proteins. For example, consider the search of ISB-All spectra against ISB+Yeast (i.e., a database formed by concatenating ISB and Yeast databases) or ISB+AT database. We do not expect any significant match between the spectra in ISB-All and proteins in Yeast or AT databases. Thus, in this case, Yeast or AT databases are dummy databases for ISB-All dataset. All PSMs matched to dummy databases are dummy PSMs.

2. Dummy spectra: The *dummy spectra* are the spectra that are not supposed to be matched to the database searched against. For instance, we sometimes appended the spectra from either AB-TC or AB-All to ISB-02, ISB-All, Y-Small, or Y-All datasets and searched the merged datasets. Since we do not expect any significant match between the spectra in AB-All or AB-TC dataset and any protein sequence database used, AB-TC or AB-All spectra are dummy spectra for all experiments. All PSMs from dummy spectra are dummy PSMs.

3. Dummy parent mass tolerance: All spectra used in our experiments were obtained with a LTQ-Orbitrap, using an MS acquisition mode where the parent mass error is usually less than 30-50 ppm . Although running database searches with parent mass tolerance 50 ppm would be enough to find most correct matches, we used 2.5 Da parent mass tolerance (*dummy tolerance*) instead.

Dummy parent mass tolerance was applied only to the experiments for the ISB-All and ISB-02 datasets. All PSMs with parent mass error larger than 50 ppm are dummy PSMs.

Note that all dummy PSMs are regarded as false but not all remaining PSMs (termed *putative* PSMs) are correct. To compute FDR (either empirical or factual) we have to estimate the number of false positive target PSMs. In case of empirical FDR, we estimate this number by the number of decoy PSMs (N_{decoy}) without distinguishing between dummy and putative PSMs. In case of factual FDR, however, we use the information that positive dummy PSMs always represent false positive PSMs; the total number of false positive PSMs is thus the number of positive dummy PSMs (denoted by N_{dummy}) plus the number of false positive PSMs among the putative PSMs (denoted by N_{false}). Since N_{dummy} is given, we only need to estimate N_{false}.

To estimate N_{false}, we use the Standard TDA Protocol. The inputs to the Standard TDA Protocol are the spectra of putative PSMs and the target database excluding any dummy proteins. The decoy database is generated by reversing this target database. For search, the parent mass tolerance is set to 50 ppm, and N_{false} is given by the number of positive decoy PSMs in this search. The factual FDR is then defined by

$$\frac{N_{dummy} + N_{false}}{N_{target}}.$$

where N_{target} denotes the number of positive target PSMs (including dummy PSMs). Since N_{false} is estimated via TDA, the factual FDR also may suffer from the bias introduced by TDA as the empirical FDR does. However, since the factual FDR is using "extra information" of dummy PSMs (not available to the database search engine nor to TDA), it is expected to be closer to true FDR than empirical FDR in particular when the number of dummy PSMs (N_{dummy}) is large. The definition of the factual FDR for two-pass searches is more complicated and is discussed below.

For each experiment, we fixed the factual/empirical FDR thresholds to 5% for searches I-1 to I-23 and 1% for searches Y-1 to Y-13 and reported the corresponding empirical/factual FDR values and the number of positive target PSMs (N_{target}). Also for each experiment, we evaluated how significant the difference between empirical FDR and factual FDR is using the Fisher's exact test [Fis]. The 2×2 tables given

in Table 4.2 were used for the Fisher's exact test. When the p-value of the Fisher's exact test (the Fisher's p-value) for a specific experiment was smaller than 5%, we regarded the empirical FDR for the experiment as inaccurate.

Note that we do not aim to compare database search engines (i.e., MS-GFDB vs. X!Tandem). We only evaluate how FDR estimation via TDA is reliable and how the number of positive PSMs (or peptides) changes for different search strategies with different parameters or protocols.

4.4 Results

4.4.1 How to construct a decoy database: reversed vs shuffled

The decoy database can be generated by reversing the target proteins (reversed), shuffling amino acids of proteins (shuffled), or enumerating amino acids randomly (randomized) [Nes10]. To avoid biased FDR estimates, it is important for decoy PSMs to have a score distribution similar to that of false target PSMs. To meet this condition, the decoy database should presumably preserve the amino acid composition (the numbers of individual amino acids) and the portion of shared peptides between different proteins in the target database. Additionally, for each spectrum, the number of target and decoy peptides matching the parent mass (within the chosen tolerance) should be similar.

The reversed database meets all these conditions when fully-tryptic peptide digestion is not enforced. Moreover, there is only one possible reversed database for every target database. This is beneficial because it removes the dependence on the randomization procedure and makes the FDR calculation deterministic and reproducible. Moreover, shuffled or randomized databases usually do not contain as many shared peptides (peptides that are shared between multiple proteins) as target protein database. This makes the actual search space in the decoy database larger than the search space in the target database, thus resulting in conservative FDR estimates [EG07]. Elias and Gygi noticed this problem and suggested a possible correction procedure [EG07] but most labs using shuffled databases still do not apply

any correction.

To assess the impact of this choice of reversed vs shuffled decoy databases, we performed various pairs of searches (Table 4.3). For each pair of searches, the search conditions differ only in the use of decoy databases - reversed or shuffled. Except the databases, all searches followed the Standard TDA Procedure. Note that we did not apply the correction suggested by Elias and Gygi in the case of shuffled database search.

Conclusion: No notable difference was observed between both approaches. Regardless of the database, both approaches reported similar numbers of PSMs at a fixed factual FDR (5% or 1%). The Fisher's p-value exceeded 5% for all cases; in contrast with popular belief, we did not observe a conservative estimation of FDR with shuffled decoy when compared to the reverse decoy database.

Based on these results, we would recommend the utilization of reversed decoy databases rather than shuffled decoy databases. While there was no noticeable disadvantage, there are several advantages of using reversed decoy databases: it is easy to generate, deterministic, reproducible and maintains the amino acid composition and distribution of shared peptides/parent masses between target and decoy databases.

4.4.2 Concatenated vs separate decoy

Given target and decoy databases, it is common to concatenate them and search the concatenated database [EG07] but some groups prefer to search them separately [KSMN08]. The difference between the two approaches is whether to allow competition between target PSMs and decoy PSMs for every spectrum. The separated search does not allow this competition in that all positive decoy PSMs are considered for FDR calculation even if the same spectra of the PSMs match to the target database with better scores.

No competition in the separated searches means rather conservative FDR estimation because the fraction of false PSMs among all target PSMs is not counted (denoted by PIT in [KSMN08], but conventionally by π_0) [KSMN08, CFN08]. Several methods to estimate π_0 were suggested (e.g., [Sto]), but we did not apply them for our experiments.

We compared both approaches using pairs of searches that differ only by the database search method - concatenated or separated search. For all searches, the standard TDA procedure was followed except step 2 for the separate search. For separate searches, the target and decoy databases are searched separately and the best scoring PSM is selected from each database and used for the empirical FDR calculation. Table 4.4 shows the results.

Conclusion:

The results show that the separate-decoy searches tend to estimate FDR conservatively. In particular for small databases, the separate-decoy searches resulted in more conservative FDR estimation than concatenated-decoy searches. For instance, the Fisher's p-value for the search I-9 was far less than 5% for both MS-GFDB and X!Tandem. This is because the π_0 factor is expected to be smaller for small databases than for large databases.

Thus, we recommend to use concatenated-decoy search. Separate-decoy searches should be used with reliable estimation of the π_0 factor, in particular for searches using small databases.

4.4.3 Choice of formula to calculate FDR

Given the numbers of target and decoy positive PSMs (denoted by N_{target} and N_{decoy} respectively), one can estimate FDR as N_{decoy}/N_{target} as in the standard TDA procedure. However, the first review on TDA by Elias and Gygi [EG07] suggested an alternative formula: $2 \cdot N_{decoy}/(N_{target} + N_{decoy})$ and both formulas are used in MS experiments (when using separate decoy with the π_0 estimation, $\pi_0 \cdot N_{decoy}/N_{target}$ should be used to estimate FDR, which is excluded because we are using concatenated decoy databases). The latter formula assumes the database search engine reports both target and decoy PSMs as positive discoveries. However, decoy PSMs do not need to be included in the final set of positive discoveries since these are obviously known to be false.

To compare how the choice of formula affects the results we modified the searches I-3, I-5, and Y-1 by changing the FDR formula (the searches I-12, I-13, and Y-4, respectively). For searches using the alternative formula, we used the second

table in Table 4.2 for the Fisher's exact test. The comparison results are shown in Table 4.5.

Conclusion: For most cases, the alternative formula ($2 \cdot N_{decoy}/(N_{target} + N_{decoy})$) resulted in conservative FDR estimation, yielding less positive target PSMs than the original formula N_{decoy}/N_{target}. For example, the Fisher's p-value was less than 5% in the search I-12 for both MS-GFDB and X!Tandem, indicating inaccurate FDR estimation from the alternative formula.

Since the FDR estimation of the original formula tends to be more accurate, we recommend using the original formula. In fact, recently Elias and Gygi also advocated using the original formula by stating that "decoy hits should not contribute to the finally of incorrect hits since they can be easily recognized and removed" [EG10].

4.4.4 Impact of the size of the database

The choice of target database is obviously critical in all MS experiments. While this database should be chosen to include the sequences of proteins contained in the sample, it should also be as compact as possible because searching a larger database takes more time and more importantly reduces the number of resulting PSMs by allowing more choices for false PSMs. The former issue is well recognized by the community but the latter is often not addressed. Since larger databases increase the chances of false matches getting high-scores, the score threshold to determine positive PSMs at a fixed FDR also becomes higher for larger databases containing higher proportions of proteins not present in the sample.

To demonstrate the effect of database size, we ran searches against various databases of different sizes and compared the results (Table 4.6).

Conclusion: As expected, for smaller databases, TDA yielded more resulting PSMs. The FDR estimation via empirical FDR was reliable regardless of the database size.

Based on these results, we recommend choosing the smallest possible database containing the sequences of proteins presumed to be in the sample.

4.4.5 How the number of spectra affects the results

In most high-throughput MS experiments, only less than 40% of all MS/MS spectra are identified. The remaining spectra are not identified because of reasons such as signal-to-noise ratio, poor peptide fragmentation, non-peptide spectra, spectra from peptides missing from the target database, post-translational modifications that are not considered in the database search, etc. If such unidentifiable spectra could be removed in advance, this would reduce the database search time and possibly produce more PSMs because unidentifiable spectra can only generate false PSMs and could thus increase the TDA-determined score threshold. To estimate the effect of unidentifiable spectra, we compared searches with various datasets differing only in the portion of unidentifiable spectra (Table 4.7).

Conclusion: Adding unidentifiable spectra reduces the number of positive PSMs, but does not change the accuracy of FDR estimations significantly. Thus filtering noisy spectra prior to a database search [BGMY04, NP06, FBS$^+$08] should be helpful towards increasing the number of resulting identifications.

4.4.6 Expected gains from accurate peptide parent masses

Modern mass spectrometry instruments (e.g., FT/ICR or Thermo LTQ Orbitrap) can measure masses very accurately and are commonly configured to generate high-accuracy MS spectra (e.g., $\leq$50 ppm) and low-accuracy MS/MS spectra (e.g., $\leq$0.5 Da) [MK08]. The availability of high-accuracy parent masses allows database search engines to greatly restrict the masses of eligible database peptides and thus significantly reduces the number of peptides scored against each spectrum. Here we measured how the availability of high-accuracy parent masses changes the results (Table 4.8).

Conclusion: As expected, when using strict parent mass tolerance more PSMs were identified (at the same factual FDR threshold) in most cases. For the searches I-17 and I-18, the empirical FDRs reported by MS-GFDB were rather inaccurate. However, while the empirical FDR in I-17 was too conservative, that in I-18 was too liberal. This indicates that the empirical FDR in searches using strict tolerance is not strongly biased toward one direction. Thus, we recommend using

strict tolerance in database searches.

4.4.7 How the score normalization affects the results

TDA implicitly assumes that given two PSMs (S_1, P_1) and (S_2, P_2) where $S_1 \neq S_2$, if $Score(S_1, P_1) \geq Score(S_2, P_2)$, the chances of (S_1, P_1) being correct should be higher than the chances of (S_2, P_2) being correct (namely, (S_1, P_1) is *better* than (S_2, P_2)). However, this is not true for all scoring functions. For example, SEQUEST Xcorr tends to assign large scores to long peptides. Thus, even if $Score(S_1, P_1) \geq Score(S_2, P_2)$, if $Length(P_1) >> Length(P_2)$, it is possible that (S_1, P_1) is a worse match than (S_2, P_2). This *score normalization* problem is an important issue for TDA to work effectively.

Using probabilistic scores (e.g. q-value, p-value or posterior error probability) is a good solution to obtain a good normalization. Most database search engines nowadays report a pair of scores: a "raw" score and a probabilistic score. For example, Mascot reports ion scores and E-values and MS-GFDB reports MS-GF score and spectral probability. Alternatively, one can get probabilistic scores by running post-processing tools like PeptideProphet [KNKA02].

To estimate the effect of score normalization, we ran pairs of MS-GFDB searches. For each pair of searches, one used the spectral probability (probabilistic score) and the other used the MS-GF score (raw score) to compute FDR. The spectral probability can be considered simply as "better normalized" score of the MS-GF score for this experiment [KGP08]. Table 4.9 shows the results.

Conclusion: Using the well-normalized score (i.e., the spectral probability) always produces substantially more resulting PSMs, with higher gains for larger databases. Furthermore, as in the search Y-7, the TDA-determined empirical FDR tended to be more accurate when well-normalized score was used. Thus, we recommend to use well-normalized scoring function (e.g., probability scores) to maximize the number of positive target PSMs at a fixed FDR. To compute FDRs separately depending on the precursor charge is also recommended if the scoring function is not well normalized across the spectra of different precursor charges. For example, most engines using peptide sequence tags (e.g., InsPecT [TSF+05]) identify spectra

of charge 2 relatively well but struggle in identifying spectra with precursor charges 3 or more. For such database search engines, it is better to compute FDRs separately depending on the precursor charge to maximize the resulting PSMs (In fact, the script to compute FDRs contained in the InsPecT package computes FDR separately for charge 2 spectra and others).

4.4.8 PSM-level vs Peptide-level FDR

In MS experiments, it is common to compute FDRs at the PSM-level (as the portion of false PSMs among positive PSMs), and use the resulting PSMs to identify peptides (if at least one PSM is identified as peptide P then P is said to be identified). These identified peptides are in turn used to identify proteins (e.g. two-peptide rule: for a protein, if it contains at least two identified peptides, it is assumed to be identified). However, while multiple correct PSMs often correspond to a single correct peptide, false PSMs typically correspond to distinct false peptides. Consequently, even a set of PSMs with a very low (PSM-level) FDR may result in excessive false peptide identifications.

Computing the empirical peptide-level FDR is a readily-available solution to this problem: if multiple PSMs are matched to the same peptide, only the best-scoring PSM is retained; the peptide-level FDR is then calculated using only these best-scoring PSMs per peptide. The factual peptide-level FDR is defined similarly.

To demonstrate the problem of PSM-level FDRs, we reported factual peptide-level FDRs for various searches when the score threshold was determined using empirical PSM-level FDR (Table 4.10). Among the searches in Table 4.10, the search I-23 illustrates the problem most explicitly. The ISB-All dataset used in the search I-23 contains spectra of 10 replicate runs of the same ISB standard protein mixture and thus many spectra are expected to be identified as the same peptides. The factual peptide-level FDRs of this search were 42.8% and 39.6% for MS-GFDB and X!Tandem, respectively.

Table 4.11 shows the comparison between empirical peptide-level FDRs and factual peptide-level FDRs. For this experiment, the Fisher's exact tests were done using number of distinct peptides instead of PSMs.

Conclusion: PSM-level FDR differs significantly from peptide-level FDRs. In particular when the larger datasets (e.g., ISB-ALL+AB-TC or Y-All+AB-TC) were used, the resulting empirical PSM-level FDRs seriously underestimated the factual peptide-level FDRs (up to 10 folds), indicating that the peptide-level FDR is more important when large datasets/experiments are considered. On the other hand, Table 4.11 demonstrates that in most cases the empirical peptide-level FDR reliably estimates the peptide-level FDR even if for some cases (e.g., the search I-23) the estimation was still too liberal.

Thus, in MS experiments where peptide identifications are used in downstream applications (e.g., protein identification) peptide-level FDR should be used instead of PSM-level FDR. Other applications choosing to use empirical PSM-level FDR should be required to present supporting evidence that such FDR estimates are accurate and appropriate for the proposed goals.

4.4.9 Two-pass searches and TDA

Craig and Beavis [CB03] pioneered the two-pass search approach that searches the target database twice. In the first pass, spectra are searched against the database to identify candidate proteins; in the second pass, spectra are again searched against only the candidate proteins identified in the first pass. The spectra matched in the first pass are sometimes removed in the second pass (matched spectrum removal (MSR) step [CB03, BK11]). This approach was originally proposed to accelerate the database search by quickly finding proteins containing non-modified fully tryptic peptide matches in the first pass and identifying more complex peptides (e.g. non-tryptic peptides or peptides with modifications) in the second pass. In addition to expediting the database search, the two-pass approach can also be used to produce more resulting PSMs by reducing the database size in the first pass.

Recently, it was recognized that TDA should be carefully applied when estimating FDRs for two-pass searches [BPG09, EBM10, BK11]. Traditionally, TDA treats a database search engine as a black box that reports a sorted list of PSMs. If we consider a database search engine supporting the two-pass search (e.g. X!Tandem [CB04]) as a black box and apply TDA, the candidate proteins selected at the first

pass will contain more target proteins than decoy proteins. Therefore, in the second pass, the assumption of TDA that matches to decoy are representative of false matches to target no longer holds and TDA will report a significantly smaller FDR than the true FDR. Results from the searches Y-10 and Y-11 in Table 4.12 illustrate this problem. When the empirical FDR was fixed to 1%, the factual FDRs of both searches were close to or exceeded 10%.

To remedy this problem, Everett et al. [EBM10] suggested to generate a decoy database for the second pass by reversing the candidate target proteins selected in the first pass. In this way, target and decoy databases in the second pass can have the same number of proteins. However, Bern and Kil [BK11] claimed that these target and decoy databases still can be "unbalanced" because the false positive PSMs in the target database are likely to have better scores than the positive decoy PSMs in the decoy database. They proposed to generate the decoy database by first taking candidate decoy protein sequences and second appending reversed sequences of candidate target protein sequences until the number of proteins in the decoy database equals to the number of the target proteins. The decoy database constructed in this way is specified by *BK decoy database*. On the other hand, the decoy database constructed by retaining only candidate decoy protein sequences is specified by *traditional decoy database*.

We tested two methods - the traditional and the BK decoy database - with or without the MSR step (we did not test the decoy database proposed by Everett et al. [EBM10]). For this experiment, only the searches using Y-All or Y-Small databases were tested because the ISB database contains too few proteins to observe the effect of the reduced target database in the second pass. From the first pass search, we used score threshold corresponding to 1% empirical FDR to find candidate proteins.

For two-pass searches, the number of dummy PSMs (N_{dummy}) can be counted as previously described (in Methods section), but the number of false positives out of putative PSMs (N_{false}) should be estimated differently because the search space of a two-pass search is typically different from a single-pass search. We call the estimation method of N_{false} for single-pass searches described in Methods section the *single-pass estimation method*.

To estimate N_{false} for two-pass searches, first consider the cases in which the

MSR step is not used. In this case, the search space is decided by the candidate proteins found in the first pass. To estimate N_{false}, for each search we take the candidate proteins found in the first pass of the search, remove dummy proteins, and generate the BK decoy database using these proteins. The spectra excluding dummy spectra are searched against the target proteins (with dummy proteins removed) and the proteins in the generated BK decoy database. N_{false} is given by the number of decoy positive PSMs in this search. This estimation method for two-pass searches is specified by the *two-pass estimation method*.

Second, in case in which the MSR step is applied, we first divide the set of spectra into two groups: S_1 matched spectra in the first step and S_2 remaining spectra. To estimate the false positives in the first pass, we use the single-pass estimation method with the spectra S_1 instead of all the spectra. To estimate the false positives in the second pass, we use the two-pass estimation method with the spectra S_2 instead of all the spectra. The final estimation of N_{false} is given by summing up the two estimated numbers of the false positives.

The results of the four two-pass search methods are shown in Table 4.12.

Conclusion: For most cases, the two-pass searches produced significantly more PSMs than the single-pass search at the same factual FDR. The empirical FDR from traditional decoy database significantly underestimated the factual FDR, in particular when the MSR step was not used (shown in the search Y-10). On the other hand, the empirical FDR from the BK decoy database was close to the factual FDR, whether the MSR step was used or not (shown in the searches Y-12 and Y-13). The numbers of target PSMs in these searches were still larger than in the single-pass search. For example, MS-GFDB reported $3262 - 2588 = 674$ and $3103 - 2588 = 515$ additional PSMs in the searches Y-12 (without the MSR step) and Y-13 (with the MSR step), respectively, as compared to the search Y-1. The factual FDRs of these additional 674 and 515 PSMs were 1.8% and 1.4%, respectively. This indicates that the additional PSMs without the MSR step result in rather high FDR.

Based on the results, we recommend to use two-pass searches using the BK decoy database because it outputs more target PSMs than single-pass searches with reliable FDR estimation.

4.5 Conclusion

Reliable estimation of false discovery rates is a necessary precondition for the downstream utility of high throughput proteomics studies. Without accurate FDR estimates it is not possible to meaningfully compare results across different labs or search procedures and substantial amounts of time and resources may be wasted following 'surprising leads' later shown to be no more than just false positives. While the final decision of which FDR (e.g., 1% or 5%) is reasonable and appropriate for a particular experiment should ultimately rest with the researcher responsible for the analysis, it is important to be aware of the expected statistical consequences of the possible procedural choices to allow for both amelioration and critical evaluation of their effects in the resulting lists of identifications. Here we evaluated these possible effects using MS/MS data from samples where we were able to define a *factual* FDR estimator of 'true' FDR using strong indicators of false identifications that were not available to TDA or the database search engine.

While the particulars of specific experiments may warrant additional exploration, the results presented here indicate that the adoption of a simple set of guidelines could substantially improve the odds that TDA estimates of 'true' FDR will be within an acceptable interval around measured empirical FDRs. Conversely, we show that there are cases where PSM-level FDR is highly inappropriate since it results in a peptide-level FDR over 10× higher than the only reported FDR. In fact, we argue that peptide-level FDR should be the norm when reporting identification results and PSM-level FDR should be avoided whenever possible and require additional evidence from the authors showing that there are substantial reasons to avoid imposing peptide-level FDR. The main reason behind this strong assertion is that most MS based experiments are conducted with the purpose of identifying peptides and proteins for biological interpretation where one is not concerned about the identity of any particular spectrum but rather with the expected number of false positives in the list of identifications used for follow up analysis. Another reasonable way to control FDR is to impose protein-level FDR; however, these procedures usually faces difficulties of their own (e.g., how to handle peptides shared by multiple proteins) and should be addressed separately in a different study. Other aspects beyond the

scope of this study that could also have a significant impact on the accuracy of TDA estimation of FDR are post-translational modifications, MS/MS acquisition modes (e.g., MS/MS + MS/MS/MS), local FDR (e.g., as used in PeptideProphet), spectral library searches [LDA10], etc.

Out of the recommendations derived and supported by the results above, we observed that two-pass searches seem to be the most promising search strategy. Out of all tested strategies, two-pass searches came closest to identifying as many peptides as would be possible with perfect advance knowledge of the exact list of proteins in the sample of interest. Of course, it should be noted that such gains are likely to deteriorate for higher complexity samples where the second pass database is not substantially smaller than the initial database. Also we remark that the increased number of identified peptides does not necessarily mean the increased number of identified proteins in two-pass searches because the candidate proteins are fixed in the first pass of the searches.

Acknowledgements

Chapter 4, in full, was published as "False discovery rates in spectral identification". K. Jeong, S. Kim, and N. Bandeira. *BMC Bioinformatics*, vol. 13(Suppl 16), S2, 11 2012. The dissertation author is the primary author of this paper.

Table 4.1: For each of the searches I-1 to I-23 (Y-1 to Y-13), we counted the numbers of positive target PSMs (N_{target}) at factual/empirical FDR 5% (1%) and computed the corresponding factual/empirical FDR of the positive PSMs. The underlined characters represent either dummy spectra, dummy databases, or dummy tolerance. [1]Search identifier; [2]MS/MS spectra used; [3]Protein database; [4]Decoy database type; [5]Reversed decoy database; [6]Shuffled decoy database; [7]Separate search against target and reversed decoy database; [8]Parent mass tolerance; [9]Dalton; [10]Parts per million; [11]Additional note; [12]Alternative formula was used to calculate FDR (see text); [13]Alternative score was used to calculate FDR (see text); [14]Two-pass searches (see text and Table 11).

Search#[1]	Spectra[2]	Database[3]	Decoy[4]	PMTol[8]	Note[11]
I-1	ISB-02	ISB	Rev[5]	2.5 Da[9]	
I-2	ISB-02	ISB	Shfl[6]	2.5 Da	
I-3	ISB-02	ISB+Yeast	Rev	2.5 Da	
I-4	ISB-02	ISB+Yeast	Shfl	2.5 Da	
I-5	ISB-02+AB-TC	ISB+Yeast	Rev	2.5 Da	
I-6	ISB-02+AB-TC	ISB+Yeast	Shfl	2.5 Da	
I-7	ISB-02+AB-TC	ISB+AT	Rev	2.5 Da	
I-8	ISB-02+AB-TC	ISB+AT	Shfl	2.5 Da	
I-9	ISB-02	ISB	Sep.Rev[7]	2.5 Da	
I-10	ISB-02+AB-TC	ISB+Yeast	Sep.Rev	2.5 Da	
I-11	ISB-02+AB-TC	ISB+AT	Sep.Rev	2.5 Da	
I-12	ISB-02	ISB+Yeast	Rev	2.5 Da	Alt.Formula[12]
I-13	ISB-02+AB-TC	ISB+Yeast	Rev	2.5 Da	Alt.Formula
I-14	ISB-02+AB-TC	ISB	Rev	2.5 Da	
I-15	ISB-02+AB-All	ISB+Yeast	Rev	2.5 Da	
I-16	ISB-02	ISB	Rev	30 ppm[10]	
I-17	ISB-02+AB-TC	ISB+Yeast	Rev	30 ppm	
I-18	ISB-02+AB-TC	ISB+AT	Rev	30 ppm	
I-19	ISB-02	ISB	Rev	2.5 Da	Alt.Score[13]
I-20	ISB-02+AB-TC	ISB+Yeast	Rev	2.5 Da	Alt.Score
I-21	ISB-02+AB-TC	ISB+AT	Rev	2.5 Da	Alt.Score
I-22	ISB-All+AB-TC	ISB+Yeast	Rev	2.5 Da	
I-23	ISB-All+AB-TC	ISB	Rev	2.5 Da	
Y-1	Y-Small+AB-TC	Yeast+AT	Rev	30 ppm	
Y-2	Y-Small+AB-TC	Yeast+AT	Shfl	30 ppm	
Y-3	Y-Small+AB-TC	Yeast+AT	Sep.Rev	30 ppm	
Y-4	Y-Small+AB-TC	Yeast+AT	Rev	30 ppm	Alt.Formula
Y-5	Y-Small+AB-TC	Yeast	Rev	30 ppm	
Y-6	Y-Small+AB-All	Yeast+AT	Rev	30 ppm	
Y-7	Y-Small+AB-TC	Yeast+AT	Rev	30 ppm	Alt.Score
Y-8	Y-All+AB-TC	Yeast+AT	Rev	30 ppm	
Y-9	Y-All+AB-TC	Yeast	Rev	30 ppm	
Y-10	Y-Small+AB-TC	Yeast+AT	Rev	30 ppm	TwoPass(1)[14]
Y-11	Y-Small+AB-TC	Yeast+AT	Rev	30 ppm	TwoPass(2)
Y-12	Y-Small+AB-TC	Yeast+AT	Rev	30 ppm	TwoPass(3)
Y-13	Y-Small+AB-TC	Yeast+AT	Rev	30 ppm	TwoPass(4)

Table 4.2: When the p-value of the Fisher's exact test (the Fisher's p-value) for a specific experiment was smaller than 5%, we regarded the empirical FDR for the experiment as inaccurate. For most searches the first table was used. The second table was used only for the searches I-12, I-13, and Y-4 (i.e., searches using the alternative formula - see Table 4.5 and text). The third table was used for experiments in Table 4.10 (empirical PSM-level FDR vs. factual peptide-level FDR), and the fourth table was for experiments in Table 4.11 (empirical peptide-level FDR vs. factual peptide-level FDR). [1]factual FDR; [2]empirical FDR; [3]factual peptide-level FDR; [4]empirical peptide-level FDR; N_{target}: the number of positive target PSMs; N_{dummy}: the number of positive dummy PSMs; N_{false}: the estimated number of false positive putative PSMs; N_{decoy}: the number of positive decoy PSMs; $N_{target\ peptides}$: the number of positive target peptides; $N_{dummy\ peptides}$: the number of positive dummy peptides; $N_{false\ peptides}$: the estimated number of false positive putative peptides; $N_{decoy\ peptides}$: the number of positive decoy peptides.

Estimator	# positives	# estimated false positives
FactFDR[1]	N_{target}	$N_{dummy} + N_{false}$
EmpiricalFDR[2]	N_{target}	N_{decoy}

Estimator	# positives	# estimated false positives
FactFDR	$N_{target} + N_{dummy} + N_{false}$	$2 \cdot N_{dummy} + N_{false}$
EmpiricalFDR	$N_{target} + N_{decoy}$	$2 \cdot N_{decoy}$

Estimator	# positives	# estimated false positives
FactPepFDR[3]	$N_{target\ peptides}$	$N_{dummy\ peptides} + N_{false\ peptides}$
EmpiricalFDR	N_{target}	N_{decoy}

Estimator	# positives	# estimated false positives
FactPepFDR	$N_{target\ peptides}$	$N_{dummy\ peptides} + N_{false\ peptides}$
EmpiricalPepFDR[4]	$N_{target\ peptides}$	$N_{decoy\ peptides}$

Table 4.3: All searches followed the standard TDA procedure except the step 2 for shuffled database searches. The results in columns labeled "FDR fixed" are obtained at empirical FDR threshold of 5% (the searches I-1 to I-8) or 1%(the searches Y-1 to Y-2). The results in columns labeled "FactFDR fixed" are obtained at factual FDR threshold of 5% (the searches I-1 to I-8) or 1%(the searches Y-1 to Y-2). The underlined characters represent either dummy spectra, dummy databases, or dummy tolerance. The first numbers in N_{target}/FactFDR/FDR/p-value fields are from MS-GFDB, and the second from X!Tandem. Note that we do not aim to compare database search engines (i.e., MS-GFDB vs. X!Tandem). We only evaluate how FDR estimation via TDA is reliable and how the number of positive PSMs (or peptides) changes for different search strategies with different parameters or protocols.
In contrast with popular belief, we did not observe a conservative estimation of FDR with shuffled decoy when compared to the reverse decoy database.
[1]the empirical FDR; [2]the factual FDR; [3] the number of positive target PSMs; [4] Fisher p-value (see Table 4.2) - Fisher p-values less than 5% were emphasized with bold fonts.

Search#	Spectra	Database	PMTol	Decoy	EmpiricalFDR[1] fixed			FactFDR[2] fixed	
					N^{3}_{target}	FactFDR(%)	p-value(%) [4]	N_{target}	EmpiricalFDR(%)
I-1	ISB-02	ISB	2.5 Da	Rev	2329/1009	5.8/4.4	10.9/30.9	2279/1024	3.9/5.7
I-2	ISB-02	ISB	2.5 Da	Shfl	2339/1023	6.0/4.6	7.2/38.6	2279/1025	3.6/5.1
I-3	ISB-02	ISB+Yeast	2.5 Da	Rev	1578/602	4.7/5.1	40.8/50.3	1583/596	5.1/3.9
I-4	ISB-02	ISB+Yeast	2.5 Da	Shfl	1597/577	5.0/4.2	50.2/34.5	1589/588	4.8/5.6
I-5	ISB-02+AB-TC	ISB+Yeast	2.5 Da	Rev	1490/569	5.0/5.8	50.2/31.2	1480/553	4.5/4.3
I-6	ISB-02+AB-TC	ISB+Yeast	2.5 Da	Shfl	1488/530	5.0/4.0	50.2/28.7	1478/550	4.9/5.5
I-7	ISB-02+AB-TC	ISB+AT	2.5 Da	Rev	1320/441	4.6/4.1	36.6/38.0	1342/464	5.8/7.3
I-8	ISB-02+AB-TC	ISB+AT	2.5 Da	Shfl	1287/441	3.4/4.1	7.5/38.0	1342/464	6.8/7.3
Y-1	Y-Small+AB-TC	Yeast+AT	30 ppm	Rev	2574/1988	1.0/1.3	50.1/22.7	2588/1759	1.0/0.5
Y-2	Y-Small+AB-TC	Yeast+AT	30 ppm	Shfl	2554/1940	0.9/1.2	38.7/27.3	2620/1758	1.1/0.6

Table 4.4: All searches followed the standard TDA procedure except the step 2 for separate reverse database searches. The search I-9 demonstrates that the separate-decoy searches result in more conservative FDR estimation than concatenated-decoy searches, in particular for small databases.

Search#	Spectra	Database	PMTol	Decoy	EmpiricalFDR fixed			FactFDR fixed	
					N_{target}	FactFDR(%)	p-value(%)	N_{target}	EmpiricalFDR(%)
I-1	ISB-02	ISB	2.5 Da	Rev	2329/1009	5.8/4.4	10.9/30.9	2279/1024	3.9/5.7
I-9	ISB-02	ISB	2.5 Da	Sep.Rev	2159/941	2.1/2.4	**0.0**/**0.4**	2287/1028	8.6/12.2
I-5	ISB-02+AB-TC	ISB+Yeast	2.5 Da	Rev	1490/569	5.0/5.8	50.2/31.2	1480/553	4.5/4.3
I-10	ISB-02+AB-TC	ISB+Yeast	2.5 Da	Sep.Rev	1462/504	4.6/3.6	40.3/18.7	1482/544	5.3/7.9
I-7	ISB-02+AB-TC	ISB+AT	2.5 Da	Rev	1320/441	4.6/4.1	36.6/38.0	1342/464	5.8/7.3
I-11	ISB-02+AB-TC	ISB+AT	2.5 Da	Sep.Rev	1287/453	3.4/4.9	**3.7**/56.1	1342/456	7.0/5.5
Y-1	Y-Small+AB-TC	Yeast+AT	30 ppm	Rev	2574/1988	1.0/1.3	50.1/22.7	2588/1759	1.0/0.5
Y-3	Y-Small+AB-TC	Yeast+AT	30 ppm	Sep.Rev	2501/1605	0.8/0.7	33.1/28.7	2589/1759	1.4/1.5

Table 4.5: The Formula field in the fifth column specifies the formula for the FDR calculation: 1 for N_{decoy}/N_{target} and 2 for $2 \cdot N_{decoy}/(N_{target} + N_{decoy})$. For all searches, the standard TDA procedure was followed except the step 4 for searches using formula 2. The searches I-12 and I-13 show that using formula 2 results in conservative FDR estimation.

Search#	Spectra	Database	PMTol	Formula	EmpiricalFDR fixed			FactFDR fixed	
					N_{target}	FactFDR(%)	p-value(%)	N_{target}	EmpiricalFDR(%)
I-3	ISB-02	ISB+Yeast	2.5 Da	1	1578/602	4.7/5.1	40.8/50.3	1583/596	5.1/3.9
I-12	ISB-02	ISB+Yeast	2.5 Da	2	1452/550	2.3/2.5	**0.0**/**2.9**	1583/596	9.6/7.4
I-5	ISB-02+AB-TC	ISB+Yeast	2.5 Da	1	1490/569	5.0/5.8	50.2/31.2	1480/553	4.5/4.3
I-13	ISB-02+AB-TC	ISB+Yeast	2.5 Da	2	1387/502	2.8/3.2	**0.6**/15.7	1480/553	8.7/8.3
Y-1	Y-Small+AB-TC	Yeast+AT	30 ppm	1	2574/1988	1.0/1.3	50.1/22.7	2588/1759	1.0/0.5
Y-4	Y-Small+AB-TC	Yeast+AT	30 ppm	2	2453/1626	0.8/0.8	27.8/36.2	2588/1759	2.0/1.0

Table 4.6: As expected, for smaller databases, TDA yielded more resulting PSMs. Fisher p-values were higher than 5% for all cases, which indicates that the FDR estimation via empirical FDR is reliable regardless of the database size.

Search#	Spectra	Database	PMTol	DB size	EmpiricalFDR fixed			FactFDR fixed	
					N_{target}	FactFDR(%)	p-value(%)	N_{target}	EmpiricalFDR(%)
I-1	ISB-02	ISB	2.5 Da	7,440	2329/1009	5.8/4.4	10.9/30.9	2279/1024	3.9/5.7
I-3	ISB-02	ISB+Yeast	2.5 Da	3,019,432	1578/602	4.7/5.1	40.8/50.3	1583/596	5.1/3.9
I-14	ISB-02+AB-TC	ISB	2.5 Da	7,440	2262/984	5.8/4.5	5.7/34.5	2221/995	4.0/6.0
I-5	ISB-02+AB-TC	ISB+Yeast	2.5 Da	3,019,432	1490/569	5.0/5.8	50.2/31.2	1480/553	4.5/4.3
I-7	ISB-02+AB-TC	ISB+AT	2.5 Da	13,475,763	1320/441	4.6/4.1	36.6/38.0	1342/464	5.8/7.3
Y-5	Y-Small+AB-TC	Yeast	30 ppm	3,011,992	3340/2734	1.2/1.0	30.0/50.1	3209/2717	0.8/1.0
Y-1	Y-Small+AB-TC	Yeast+AT	30 ppm	16,480,315	2574/1988	1.0/1.3	50.1/22.7	2588/1759	1.0/0.5

Table 4.7: Adding unidentifiable spectra reduces the number of positive PSMs, but does not change the accuracy of FDR estimations significantly.

Search#	Spectra	Database	PMTol	# spec	EmpiricalFDR fixed			FactFDR fixed	
					N_{target}	FactFDR(%)	p-value(%)	N_{target}	EmpiricalFDR(%)
I-1	ISB-02	ISB	2.5 Da	4,966	2329/1009	5.8/4.4	10.9/30.9	2279/1024	3.9/5.7
I-14	ISB-02+AB-TC	ISB	2.5 Da	11,285	2262/984	5.8/4.5	5.7/34.5	2221/995	4.0/6.0
I-3	ISB-02	ISB+Yeast	2.5 Da	4,966	1578/602	4.7/5.1	40.8/50.3	1583/596	5.1/3.9
I-5	ISB-02+AB-TC	ISB+Yeast	2.5 Da	11,285	1490/569	5.0/5.8	50.2/31.2	1480/553	4.5/4.3
I-15	ISB-02+AB-All	ISB+Yeast	2.5 Da	24,948	1367/531	4.2/5.3	33.0/34.5	1393/518	5.5/4.2
Y-1	Y-Small+AB-TC	Yeast+AT	30 ppm	16,077	2574/1988	1.0/1.3	50.1/22.7	2588/1759	1.0/0.5
Y-6	Y-Small+AB-All	Yeast+AT	30 ppm	29,740	2238/1913	1.1/1.4	44.2/14.7	2208/1629	0.9/0.7

Table 4.8: As expected, when using strict parent mass tolerance more PSMs were identified (at the same factual FDR threshold) in most cases.
[1]For the search I-16, the factual FDR is not available because no dummy element is used.

Search#	Spectra	Database	PMTol	EmpiricalFDR fixed			FactFDR fixed	
				N_{target}	FactFDR(%)	p-value(%)	N_{target}	EmpiricalFDR(%)
I-1	ISB-02	ISB	2.5 Da	2329/1009	5.8/4.4	10.9/30.9	2279/1024	3.9/5.7
I-16	ISB-02	ISB	30 ppm	2128/1009	N/A[1]	N/A	N/A	N/A
I-5	ISB-02+AB-TC	ISB+Yeast	2.5 Da	1490/569	5.0/5.8	50.2/31.2	1480/553	4.5/4.3
I-17	ISB-02+AB-TC	ISB+Yeast	30 ppm	1638/569	6.4/5.3	**3.1**/45.2	1570/565	3.9/4.8
I-7	ISB-02+AB-TC	ISB+AT	2.5 Da	1320/441	4.6/4.1	36.6/38.0	1342/464	5.8/7.3
I-18	ISB-02+AB-TC	ISB+AT	30 ppm	1358/463	3.2/5.1	**2.0**/50.3	1425/463	6.5/4.3

Table 4.9: The spectral probability can be considered simply as "better normalized" score of the MS-GF score for this experiment [KGP08]. Using the well-normalized score (i.e., the spectral probability) always produces substantially more resulting PSMs, with higher gains for larger databases. Furthermore, as in the search Y-7, the TDA-determined empirical FDR tended to be more accurate when well-normalized score was used.
[1]Spectral probability was used to compute the FDR; [2]MS-GF score was used to compute the FDR.

Search#	Spectra	Database	PMTol	Score	EmpiricalFDR fixed			FactFDR fixed	
					N_{target}	FactFDR(%)	p-value(%)	N_{target}	EmpiricalFDR(%)
I-1	ISB-02	ISB	2.5 Da	SpecProb[1]	2329	5.8	10.9	2279	4.4
I-19	ISB-02	ISB	2.5 Da	MSGFRaw[2]	2079	4.6	36.5	2079	4.6
I-5	ISB-02+AB-TC	ISB+Yeast	2.5 Da	SpecProb	1490	5.0	50.2	1480	4.5
I-20	ISB-02+AB-TC	ISB+Yeast	2.5 Da	MSGFRaw	1272	5.7	25.2	1210	4.5
I-7	ISB-02+AB-TC	ISB+AT	2.5 Da	SpecProb	1320	4.6	36.6	1342	5.8
I-21	ISB-02+AB-TC	ISB+AT	2.5 Da	MSGFRaw	987	3.9	37.3	1064	6.1
Y-1	Y-Small+AB-TC	Yeast+AT	30 ppm	SpecProb	2574	1.0	50.1	2588	1.0
Y-7	Y-Small+AB-TC	Yeast+AT	30 ppm	MSGFRaw	1215	1.9	**1.8**	861	0.3

Table 4.10: Score thresholds were determined using PSM-level FDR thresholds and used to calculate factual peptide-level FDRs. The results illustrate that PSM-level empirical FDR underestimates peptide-level FDR significantly (e.g., the searches I-22 and I-23).
[1]Number of distinct peptides; [2]Factual peptide-level empirical FDR.

Search#	Spectra	Database	PMTol	EmpiricalFDR fixed			
				N_{target}	# peptides[1]	FactPepFDR(%)[2]	p-value(%)
I-5	ISB-02+AB-TC	ISB+Yeast	2.5 Da	1490/569	600/262	12.0/11.8	**0.0**/**0.1**
I-22	ISB-All+AB-TC	ISB+Yeast	2.5 Da	13441/5086	1375/538	38.6/38.1	**0.0**/**0.0**
I-14	ISB-02+AB-TC	ISB	2.5 Da	2262/984	815/361	13.1/10.5	**0.0**/**0.1**
I-23	ISB-All+AB-TC	ISB	2.5 Da	19501/8497	1628/556	42.8/39.6	**0.0**/**0.0**
Y-1	Y-Small+AB-TC	Yeast+AT	30 ppm	2574/1988	2355/1841	1.1/1.4	37.8/15.8
Y-8	Y-All+AB-TC	Yeast+AT	30 ppm	9005/6269	3567/2640	2.4/2.0	**0.0**/**0.0**
Y-5	Y-Small+AB-TC	Yeast	30 ppm	3340/2734	3033/2515	1.2/1.0	19.1/54.5
Y-9	Y-All+AB-TC	Yeast	30 ppm	11151/8969	4341/3582	2.3/2.3	**0.0**/**0.0**

Table 4.11: Score thresholds were determined using empirical/factual peptide-level FDR and used to calculate factual/empirical FDRs. For the searches I-5, I-22, I-14, and I-23, the peptide-level FDR thresholds were set to 5%, and for the remaining searches they were set to 1%. The search I-23 illustrates the difficulty of enforcing peptide-level FDR when searching small databases.
[1]Empirical peptide-level FDR.

Search#	Spectra	Database	PMTol	EmpiricalPepFDR[1] fixed				FactPepFDR fixed		
				N_{target}	# peptides	FactPepFDR(%)	p-value(%)	N_{target}	# peptides	EmpiricalPepFDR(%)
I-5	ISB-02+AB-TC	ISB+Yeast	2.5 Da	1340/498	532/234	5.3/6.4	45.0/22.6	1335/479	529/226	4.9/4.4
I-22	ISB-All+AB-TC	ISB+Yeast	2.5 Da	10245/3688	758/304	6.6/6.6	7.0/**4.3**	9448/3596	696/292	2.4/4.1
I-14	ISB-02+AB-TC	ISB	2.5 Da	2088/907	727/333	6.7/5.1	10.4/50.3	1994/900	693/332	2.9/4.8
I-23	ISB-All+AB-TC	ISB	2.5 Da	16602/6676	1083/416	9.2/8.9	**0.0**/**2.0**	15663/6203	1015/385	1.7/1.3
Y-1	Y-Small+AB-TC	Yeast+AT	30 ppm	2574/1963	2355/1818	1.1/1.3	38.7/17.6	2556/1759	2339/1636	0.9/0.6
Y-8	Y-All+AB-TC	Yeast+AT	30 ppm	7867/5068	3142/2201	1.3/1.0	6.1/50.1	7121/5068	2849/2201	0.4/1.0
Y-5	Y-Small+AB-TC	Yeast	30 ppm	3209/2666	2916/2455	1.0/0.9	44.7/50.1	3209/2734	2916/2515	0.9/1.0
Y-9	Y-All+AB-TC	Yeast	30 ppm	10005/7309	3885/2987	1.0/0.9	50.0/44.9	10005/7309	3885/2987	0.9/0.9

Table 4.12: For the searches Y-10 and Y-11, the traditional second pass decoy database was used to estimate FDR (see text). For the searches Y-12 and Y-13, the decoy database proposed by Bern et al. [BK11] was used. Also, for the searches Y-11 and Y-13, the matched spectrum removal (MSR) step was used.
Low Fisher p-values in Y-10 and Y-11 illustrate that using the traditional second pass decoy database results in significant underestimation of the true FDR.
[1]The decoy database for the second pass search; [2]Whether the matched spectrum removal step was used; [3]The traditional decoy database; [4]The BK decoy database.

Search#	Spectra	Database	PMTol	IsTwoPass	2th decoy[1]	MSR[2]	EmpiricalFDR fixed			FactFDR fixed	
							N_{target}	FactFDR(%)	p-vale(%)	N_{target}	EmpiricalFDR(%)
Y-1	Y-Small+AB-TC	Yeast+AT	30 ppm	No	Rev	N/A	2574/1988	1.0/1.3	50.1/22.7	2588/1759	1.0/0.5
Y-10	Y-Small+AB-TC	Yeast+AT	30 ppm	Yes	Trad[3]	No	5361/5744	15.9/20.1	**0.0/0.0**	3260/2655	0.6/0.3
Y-11	Y-Small+AB-TC	Yeast+AT	30 ppm	Yes	Trad	Yes	4114/3925	7.3/10.1	**0.0/0.0**	3102/2320	0.9/0.5
Y-12	Y-Small+AB-TC	Yeast+AT	30 ppm	Yes	BK[4]	No	3529/3089	1.2/1.0	24.9/45.0	3262/3074	0.7/0.9
Y-13	Y-Small+AB-TC	Yeast+AT	30 ppm	Yes	BK	Yes	3137/2514	1.1/1.1	40.1/39.3	3103/2521	1.0/0.8

Chapter 5

Virmid: virtual microdissection of sample mixtures for accurate somatic mutation profiling

5.1 Introduction

Identifying mutations relevant to a specific phenotype is one of the primary goals in sequence analysis. With the advent of massively parallel sequencing technologies, we can produce an immense amount of genomic information to estimate the landscape of sequence variations. However, the error rate of base-call and read alignment still remains much higher than the empirical frequencies of single nucleotide variations (SNVs) and *de novo* mutations[SMG12a]. Many statistical methods have been proposed to strengthen mutation discovery in the presence of confounding errors[LHW$^+$09, GSM$^+$10, DBP$^+$11].

Finding somatic mutations is one particular type of variant calling, which constitutes an essential step of clinical genotyping. Unlike the procedures used for germline mutation discovery, the availability of matched control sample is indispensable. Here, sequence variants that exist in the control sample are used as a basis for measuring individual polymorphisms, while the disease-only mutations are generally regarded as candidate somatic mutations. Traditional approaches call variants from each sample to estimate the sequential differences[KCW$^+$09, LHC$^+$12]. But most

recent studies that calculate joint probabilities of the disease-control genotype pairs showed higher efficiency in separating true somatic mutations from germline mutations by considering correlations between two samples [KZL+12, RMD+12, SWS+12]. With the aid of probabilistic variant calling models, whole genome/exome sequencing data have been used to identify potential *de novo* mutations in various studies including schizophrenia[XRD+11], autism[SMG+12b], and cancer[BBL+12].

However, there are many cases where mutation discovery might be confounded. One big hurdle is the impurity and heterogeneity of the disease sample. For example, gastric and breast cancer tissues usually contain large amount of stromal cells to make the acquisition of pure cancer sample not feasible[MUD+08]. More importantly, there are many cases in which this type of contamination is not only inevitable but dominating the sample constitution. Focal malformation of cortical developments including focal cortical dysplasia and hemimegalencephaly is the most common cause of childhood intractable epilepsy and contain diseased cells in affected brain regions with high proportion of normal cells [LHS+12]. Similar problem arises when detecting small amount of target genome mixed in the control samples. In organ transplant, an increased level of circulating cell free DNAs ($\sim$10%) of the donor in the recipient's blood indicates a higher risk of failure[SKVQ11]. Cell-free DNAs are also found in pregnancy; small amount of fetal DNAs ($\sim$13%) are detectable in maternal plasma[KSV+12]. In both cases, accurate identification of the target genotypes will provide the basis for a non-invasive and low cost diagnostic method.

Conventional methods for somatic mutation profiling are severely compromised in highly contaminated samples because the abundant short reads originated from control genomes obscure true allele frequency (AF) at the site of *de novo* mutations. This usually results in a failure to call true variants. We have two questions that arise: (a) estimate the contamination level, defined as the proportion of control sample in the mixed disease sample (α, $0 \leq \alpha \leq 1$) and (b) use α in SNV calling. A natural approach to estimating α has been adapted by many previous studies[SKVQ11, YMJ+10, CMF+11, CCH+12, SZZ+12]. For any heterozygous mutation, the `B` allele frequency (BAF) is expected to be close to 50%. A significant and consistent deviation from this value is indicative of the existence and level of control sample inclusion. We found, however, there are two substantial problems in

this approach. First, it needs an initial SNV calling procedure either from sequencing or SNP array data, which takes extra time and cost. Second, and more importantly, the initial mutation call is not representative; higher BAF is likely to be observed in the selected sites causing underestimation of α. We will show that the bias is significantly large in highly contaminated samples and describe the way to resolve it. Incorporating the estimated α in SNV calling model is another important problem. There are only a few studies that consider α or similar concept in SNV calling [KZL$^+$12, SWS$^+$12]. More rigorous and explicit use of α by a tight parameterization in the probabilistic model will improve the accuracy of final calls.

Here, we describe a novel probabilistic model Virmid (**vir**tual **mi**cro**d**issection for variant calling) which estimates 1) the sample contamination level (α), and 2) the disease genotypes including somatic mutations (Figure 5.1). In the core of Virmid lies a maximum likelihood estimator (MLE) of α and the joint probabilities of control and disease genotypes represented by *joint genotype probability matrix* $\mathcal{G}$ (see Methods), driven from a local distribution of BAF. It does not require any prior SNV calling nor does it attempt to find variants beforehand; we show that this not only saves computation but also greatly improves the accuracy by reducing sampling bias. Our model also incorporates other sources of noise including sequencing error, mapping error, read mapping bias and mappability[LS12] of the genomic regions for more accurate modeling, as well as the effect of copy number variations. More importantly, the tight coupling of α and $\mathcal{G}$ implemented in a single integrated model enables rigorous recalibration of genotype probabilities with the given α. We demonstrate on simulated and real exome sequencing data that Virmid significantly increases overall precision and recall in variant finding. Even in some intractable cases, where the target genome exists in a very small amount ($\alpha \geq 80\%$) in the sample, Virmid shows a near-robust performance. We expect that this improvement will contribute to many related problems from cancer somatic mutation profiling to contaminant genome identification.

5.2 Results and Discussions

Virmid workflow: The Virmid workflow is shown in Figure 5.1. The input to Virmid includes short reads sequenced from a pure control sample and a potentially mixed disease sample. As a pre-processing step, the reads were aligned to the reference genome to generate sequence alignments. Second, the alignments were corrected using post-processing tools such as GATK's IndelRealigner [MHB+10]. Third, BAF was calculated from the corrected alignments for every nucleotide position. Fourth, initial filters were applied for quality control as well as reduction in sample size. Due to the large size of usual genomic data, we implemented a multi-tier sampling strategy (Figure 5.2), which reduced overall running time and disk usage about 7 to 10 fold. Finally, two filtered alignment files (pileup format) from control and disease sample were prepared as input.

The first step for Virmid is the estimation of α. Here we denote `A` for the reference allele, and `B` for non-reference. The set of diploid genotypes is, thus, given by $G = \{\texttt{AA},\texttt{AB},\texttt{BB}\}$. As α is a global parameter that affects all positions equally, a small subset of positions is sufficient. To obtain robust and unbiased estimates, we used a number of criteria. 1) We used only the positions with no `B` allele observed in controls to maximize the chance of getting true somatic mutations; 2) we eliminated positions with very high or low coverage suggestive of CNV; 3) more filters were applied so that the selected positions had mapping and sequencing quality values above a certain threshold; and, 4) the known mappability [LS12] of the corresponding reference region had to be above a certain threshold (see Methods for detailed setting of Virmid). Finally, the sites were filtered to remove alleles with BAF lower than a parameter R ($0 < R \leq 1$). While this makes the filtered list biased for higher BAF mutations, the explicit parameter value R was incorporated in our model to correct that bias (see Methods).

Virmid estimated α from the sampled sites using a Maximum Likelihood Estimator (MLE) [Kay93] with gradient descent search, and simultaneously estimated a joint genotype probability matrix $\mathcal{G}$, based on the estimated α. The estimated α value and the matrix $\mathcal{G}$ were used to call the most likely genotype at every nucleotide position. Finally, somatic mutation filters were applied to reduce false positives (see

Methods). The overall pipeline including data preprocessing is implemented as a single Java program. We utilized open source libraries such as samtools and picard to increase efficiency and compatibility of the program. We could also significantly reduce the use of memory and disk space using tabix [Li11]. Once pileup files were generated, the running time for an typical whole exome data (40x) was less than 2 CPU hours (Intel i7-2600 processor).

5.2.1 Test on Simulated data

Simulated control and disease genomes were prepared from human chromosome 1 (hg19) by introducing random mutations. Out of 275,814 germline (mutation rate: 10^{-3}) and 2,522 somatic mutations (mutation rate: 10^{-5}), 47,796 and 257 mutations were located in non-detectable regions (e.g. reference genotype is unavailable) leaving only 228,018 and 2,265 mutations as a true answer set. Disease samples were generated by artificially mixing two genomes in 11 different portions (α = 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90%). The Illumina-like short reads (read length=100 bp) in a medium (40x) coverage were mapped to the reference and used as the input of the Virmid pipeline (see Methods for the complete protocol).

Contamination level estimation

The estimated sample contamination levels on the 11 different mixtures are shown in Figure 5.3 (red line with circles) and Table 5.1. The overall accuracy was near perfect with only 0.53% mean deviation from the true value. To test robustness, we ran Virmid 20 times for each data varying the sampling parameter R (minimum BAF to control the number of sampling points). All replicated results were bounded within 2% ($0.19 \leq st.dev. \leq 0.53$) showing that the estimation with MLE is robust (Table 5.1). We found there is only a minor ($\pm \sim 1$) overestimation in very lowly ($\alpha \leq 5\%$) and underestimation in very highly ($\alpha \geq 80\%$) contaminated samples. However the error size was negligible compared with a conventional call-based calculation (see Methods), which estimates α based on initially identified somatic mutations (Figure 5.3, green line with circles).

We note two types of biases in the call-based method (see Figure 5.3, green

line) , loss of reads (LOR) and loss or variants (LOV) that lead to overestimation and underestimation of α respectively. LOR originates from the difference of mappability among short reads at the site of somatic mutation. Assume a disease genome has a heterozygous somatic mutation (`AB`) at position i. As the reference genome has an `A` genotype, reads with `A` at position i are more likely to be mapped. This results in underrepresentation of `B` allele, followed by an overestimation of α (see Methods for details). LOV is caused from the tendency that the variant calling is more favorable to the regions of higher BAF. Assume that a disease sample of α contamination has `AB` heterozygous mutations. In these positions, BAF follows a binomial (or similar) distribution with a probability of choosing `B` allele of $(1 - \alpha)/2$. In conventional SNV calling algorithms, the positions with higher BAF are easier to be discovered. Therefore, the distribution of BAF of the *called* mutations is shifted upward, which results in overrepresentation of `B` allele, followed by underestimation of α. The effects of the two estimation biases are dependent on α. The difference in the number of mapped reads with `A` and `B` allele is proportional to the absolute number of reads generated from disease genome. So, the LOR bias is inversely proportional to α. On the other hand, LOV effect is proportional to α because the SNV calling performance remains robust in the low contamination samples. The combined effect explains the bimodal error distribution of call based method. Eventually, the estimation result shows the suggested biases exist and are corrected efficiently in Virmid (see Methods).

Because we do not rely on initial mutation calling, the sites used for α estimation may contain non-mutated positions. As we already filtered out all the positions where control sample has one or more `B` alleles, only three possible joint genotypes remain: (`AA-AA`, `AA-AB`, and `AA-BB`). Thus, Virmid estimates the frequencies of these genotypes along with α. Since the likelihood we used is dependent on α and the frequencies of the genotypes, we attempt to find the combination of α and the genotype frequencies that maximizes the likelihood. We showed empirically that the likelihood space is convex and maximized near the true answers (Figure 5.4 A). Therefore, we used a fast gradient descent search algorithm to get MLE estimates, instead of slower EM like algorithms (Figure 5.4 B, see Methods).

Somatic mutation calling

We ran Virmid to predict the most probable genotype for each nucleotide position in the simulated data set using the estimated α. Somatic mutations were called based on the predicted genotype probabilities after filtering. To evaluate the influence of α, we compared the result with those from other SNV calling tools including JointSNVMix2, Strelka and VarScan2 (Figure 5.5). Virmid and VarScan2 can take tumor purity and ran in two different modes (with and without α); note that VarScan2 does not estimate α, and was provided with our estimation. Strelka generates two outputs, a standard and a filtered mutation lists (see Methods for detailed protocol). Evaluation was done against the 2,265 true somatic mutations cataloged from simulation procedure based on precision-recall curves (Figure 5.5 A) where exact genotype probabilities are available (Virmid and JointSNVMix2), or a single precision, recall, and f-score (Figure 5.5 B) where only final mutation lists are provided (Strelka and VarScan2).

The performance of all algorithms was comparable for relatively low contamination ($\alpha \leq 50$), but varied considerably for higher α values. Generally, tools that incorporate contamination level (Virmid w/α, VarScan2 w/α, Strelka; Strelka has a non-explicit noise level that may indicate tumor purity) outperformed the ones that did not (Virmid w/o α, VarScan2 w/o α, JointSNVMix2). The point is clearer when the same tool with different α parameters are compared (Virmid and VarScan2 w/$\alpha \leftrightarrow$w/o α).

A detailed analysis on the BAF distribution in different call sets provides a second test of performance (Figure 5.6). Note that the mean BAF is given by $(1-\alpha)/2$. As expected, the BAF distribution of the true mutation set (Figure 5.6, pink bar) decreases as α increases; With low α, there is no major problem in detecting somatic mutations because BAF is high enough to be distinguished from non-mutational sequencing/mapping error frequencies. However, in cases with high α, algorithms start to fail in calling somatic mutations with relatively low BAF. The ultimate case is that there is no `B` allele observed in disease sample due to the low portion of true disease genome and variance. For example, 316 out of 2,265 somatic mutations sites had no reads with `B` allele in 90% α sample. As there is absolutely no way to detect these

sites, the called mutation set must have higher BAF distribution.

Finally, we revisit the coverage issue in SNV calling. Although moderate (40x) coverage is generally considered sufficient for SNV calling (calculated from [SASK12]), high contamination needs higher coverage. For example, with 90% contamination, only 5% of reads (or 2 reads in 40x coverage) will sample `B` allele. Higher coverage adds more confidence to each position's genotype probability by providing more reads to observe. To see the effect of higher coverage, we generated 100x simulation data sets from three highly contaminated data (α=70%, 80%, and 90%). The data sets were analyzed using Virmid. Table 5.2 shows the improvement of prediction performance (especially in recall). With 80% contamination, Virmid could identify 94% of the true somatic mutations with almost perfect accuracy. Even with 90% contamination, 68% of true somatic mutations could be discovered, which is improved by more than 250% (611 to 1545) from 40x coverage result with better precision (0.96 to 0.98). From the result, we can conclude that deeper coverage 1) greatly improves the mutation finding in highly contaminated samples and 2) should be considered when sample purity is questionable.

Although testing on simulated data gives a unique benefit to test exact precision and recall by proving true answer set, there is some limitation too. Many difficulties in somatic mutation detection come from ambiguous read mapping. In simulation, the same reference genome assembly is used in artificial read generation. However, in real data, donor genomes contains significantly more variations other than SNVs such as copy number variations and structural variations [LSN+07]. Therefore, we proceed to test on publicly available disease data for more extensive validation of Virmid's performance.

5.2.2 Test on breast cancer data

To test with real disease data, we ran Virmid on 15 whole exome breast cancer dataset generated from The Cancer Genome Atlas (TCGA) project [Can12] (see Table 5.3). Breast cancer is known to contain high amount of stromal cells in the tumor mass[NZVLW+12], which makes the relevant genetic studies more challenging. In this context, estimating and considering the level of impurity might be necessary

for more accurate analysis of somatic mutation finding.

Before reporting accuracy, exact meaning of sensitivity and specificity of the test should be defined. Note that in the absence of a complete list of true positives, predicted but not confirmed calls cannot be treated as false positives. To test specificity, we generate false tumor/normal pairs from the same sample, where every call is a false positive. We applied a "virtual tumor" approach, suggested by MuTect study [CLC+13] for the specificity test.

We first measured the sample impurity by estimating α (Figure 5.7 A, blue area). The values were ranged from 0.41 to 0.77. Unlike other monoclonal disease, we note that there is a chance of overestimation due to the genetic heterogeneity in cancer (independently addressed in other study [CCH+12]). However, we found that the impurity range is generally consistent with the previous measurement from 21 breast cancers [NZVLW+12] and with TCGA's sample quality control step (see Methods).

There were 1654 experimentally validated mutations in the 15 exome data (see Methods). Figure 5.7 A shows the measured sensitivity from four different callers (Virmid, Strelka, JointSNVMix and MuTect); we excluded VarScan2 because it has been used to generate the initial TCGA callset. The recent JointSNVMix contains its own filtering module. We found the filtered version of JointSNVMix is always better than unfiltered by not losing a single true answer. So, we will only use the filtered version for JointSNVMix for comparison. For Strelka, we found that the default Strelka filter eliminates most of the true answers (see Methods). So, we manually disabled its read depth filter (DP) that is too strict to exome sequencing data to get the best filtered result of Strelka. Out of 1654, we only missed 23 validated mutations (36 in unfiltered Strelka, 47 in MuTect, 95 in filtered Strelka, 255 in JointSNVMix) to mark the best sensitivity 98.61% (97.82% in unfiltered Strelka, 97.16% in MuTect, 94.26% in filtered Strelka, 84.58% in JointSNVMix). Ordered by pre-calculated α, we found a clear decrease of sensitivity in JointSNVMix, which does not estimate sample impurity (Figure 5.7 A, black trend line). The relatively smaller sensitivity increase compared to Strelka and MuTect, which are shown to be consistent in the given α range, can be explained by other features such as unbiased estimation of sample purity and more rigorous filtering.

We then analyzed the types of negative calls. Out of 47, 95 and 255 false negative calls from filtered Strelka, MuTect and JointSNVMix, Virmid recovered 32, 81, and 252 mutations, which corresponds to 68.1%, 85.3%, and 98.8% of each false negatives. The read depths (of normal and tumor sample) and allele frequencies of the recovered mutations are shown in Figure 5.7 B. We found the main reason of missing calls in JointSNVMix is low allele frequency (yellow dots in Figure 5.7 B upper, $\mu = 13.3\%$), while many from Strelka and MuTect resulted from low read depth (blue and purple dots, median read depth=18). Because the subtle changes in genotype probability more critically affect in lower read depth, this results also proves the reliability of Virmid's genotype calculation model. We also analyzed the 23 false negatives of Virmid (Figure 5.7 B lower). In most case (15/23), the read depth in normal sample was extremely low (red dots in Figure 5.7 B lower, median read depth=7). We found Virmid called these mutations as germline mutations (`AB-AB`). Although missed in consequence, we are convinced our call is not wrong in theory, because in low read depth, the probability of having only reference sequences from `AB` genotype (just by chance, calculated from binomial distribution) is much higher than the prior probability of having somatic mutation (`AA-AB`). The only solution for this, is increasing (e.g. by balancing out) read depth in normal sample, because calling these regions will greatly increase false positive rate. Overall, the false negatives from Virmid were partially recovered by other tools (8, 9, and 20 by Strelka, MuTect and JointSNVMix).

In this test, sensitivity increases monotonically according to the total number of calls. It is informative to compare the number of calls to achieve a similar level of sensitivity. Figure 5.7 C shows the total number of predicted mutations. We found that unfiltered Strelka ($\sim$ 5646 per sample) and JointSNVMix ($\sim$ 7362 per sample) predicted far more mutations than others (941, 336 and 738 per sample in Virmid, filtered Strelka, and MuTect respectively). Although we cannot assert non-validated calls are all false, we can suspect more false positive calls in those two tools.

To ensure the specificity, we designed "virtual tumor" by dividing high depth (> 80) normal samples into two artificial samples including one $faked$ tumor and one $faked$ control sample. Because all the reads are originally generated from the same genome, any positive calls on these samples can be considered as false positives. We

ran all tools on five "virtual tumor" sets with same parameters used in breast cancer data to get estimated false positive rates (Figure 5.7 D). Filtered Strelka showed a surprising specificity (< 0.01 false calls per million base-pair (Mbp); note that the sensitivity was limited. Virmid and MuTect also showed a satisfactory performance ($\sim$ 1 per Mbp). Unfiltered Strelka, which showed almost comparable sensitivity in breast cancer data with Virmid, however, contained more false positives ($\sim$ 3 per Mbp). JointSNVMix, even after applying its own filtering method, is shown to contain more false positives than other tools.

We note that on simulated data, Virmid's performance compared to other tools was similar for samples with up to 50% contaimnation, but becomes progressively better for higher contamination levels such as those exceeding 70%. The experimental data presented here is at mid levels of contamination (41% to 77%), which is not ideal to showcase Virmid's strengths. In the next section, we also discuss a new data set with higher levels of contamination, but without independently validated mutations. As validated data sets grow in number, the advantages of calling mutations after estimating α will be more apparent.

5.2.3 Application to HME exome sequencing data

We applied Virmid to the recently sequenced disease/normal paired data of five hemimegalencephaly (HME) patients[LHS+12]. HME is a rare disease characterized by the enlargement and malformation in one cerebral hemisphere and is known to be an important cause of epilepsy and developmental delay. One distinctive histopathological feature of HME is that the dysmorphic and immature neurons are dispersed in the disease lesion. In this condition, the brain samples from surgical resection are expected to contain significant amount of non-disease cells. Also, the mutation burden measured by whole exome sequencing and mass spectrometry from three previously validated mutations (AKT3 c.49C>T in HME-1565, MTOR c.4448C>T in HME-1563, and PIK3CA c.1633G>A in HME-1573) assured the compromised sample purity (see Table 5.4). The dropped allele frequencies (9.7%$\sim$30.38%) are far less than the expected (50% for heterozygous, 100% for homozygous mutation sites) indicating the existence of reference alleles (`AA`) from normal cells. As the mutations

are believed to occur during early cerebral development and the surgical treatment is done in infants, the low mutation burden is less explained by disease subpopulation.

To estimate the sample contamination level, we ran Virmid 20 times for each whole exome sequencing data with variating sampling parameters (see Methods). The estimated α for the five samples (HME-1563, HME-1565, HME-1573, HME-1574 and HME-1620) are surprisingly high ranging from 64.0% to 84.8% (Figure 5.7 A), which indicates that only 15.2% to 36% of the sample is disease cell. The low variance ($<$0.075) within the same sample gives high confidence to the estimated values. While our manuscript was under review process, an independent study [ECL$^+$12] measured mutation burden of the same disease using 100 single cell sequencing, which reported a consistent result (39% in NeuN$^+$ and 27% in NeuN$^-$ populations). We also checked the distribution of BAF at the site of candidate somatic mutations (Figure 5.7 B). Note the overall drop of BAF towards zero as shown in the high α simulated examples (Figure 5.6 right). The average BAF was perfectly negatively correlated with the expected BAF calculated from $(1-\alpha)/2$. For example, the sample HME-1573, which has been predicted to have lowest contamination (α=64.0%), had the highest BAF distribution. This negative correlation is consistent with our assumption that higher α leads to lower BAF.

Although there is no efficient way to measure the true contamination levels in the sample, the allele frequencies of validated mutations (AKT3, MTOR and PIK3CA) provide a good reference. In HME-1563 and HME-1573, the validated BAF values (Figure 5.7 B, red triangles) were very close to the expected heterozygous BAF. In HME-1565, the validated BAF (28%) is twice as the expected BAF. We checked the genotype probability of the corresponding mutation and found that its probability of being homozygous is significantly high (7.1%, ranked 2nd out of 496 mutations). In all cases, mutation burdens measured from peak intensity of mass spectrometry (Figure 5.7 B, blue squares) are also bounded in a low BAF range (8%$\sim$40%) Although the allele frequencies at a single position is still highly variable to be a sample level estimator, we are convinced that the HME samples contain large amount of normal cells and the estimated α are reasonably ranged by aggregating various inspections. Identifying and quantifying more somatic mutations will lead to better validation of the sample contamination level.

Finally, we ran Virmid on the same dataset to find novel somatic mutations that might have been missed due to the sample impurity. Virmid predicted totally 2,787 somatic mutations from five individual data sets, only 653 (23.4%) of which were predicted from the previous finding. (see Table 5.5). Note that the number of newly predicted mutations is correlated with the estimated α; the higher the α is, the more somatic mutations could be missed from conventional approaches. Out of 2,134 newly predicted sites, 1,377 were located in exon region containing 923 missense events. As we expected, Virmid successfully discovered all the previously reported *de novo* mutations of PIK3CA, AKT3, and MTOR with very high confidence ($p \sim 1.0$). We focused on the two (HME-1574 and HME-1620) samples where no meaningful somatic missense mutations were not detected in the previous study. In HME-1574, Virmid discovered a novel somatic mutation of MTOR (MTOR p.Ala1517Thr) located nearby one of the validated mutations (MTOR p.Cys1483Tyr). At this site, only 3 out of 54 mapped reads represented `B` allele (BAF=5.56%) while no `B` allele was found in the control sample; we could not find any sign of mapping ambiguity nor compromised base call quality. We are convinced that the extremely reduced BAF is the major reason of the unsuccessful finding in conventional approaches. A functional analysis showed that the region is highly conserved (phastCons [SBP$^+$05] score=1.0) and no other mutation has been known at the same site so far. We expect a further experimental validation can verify the mutation in other patients as well as the more accurate effects on the protein activity (e.g. gain of function). Virmid also detected more candidates of *de novo* somatic mutations in HME-1620, but we could not find meaningful missense mutations directly linked to the PI3K-ATK-mTOR pathway or HME pathogenesis. It is possibly due to either that disease-causing mutations exist other than in coding regions or that the combinatorial effect of the low coverage of current exome sequencing (∼30x) and the high α (84.8%). As we showed in the simulation (Table 5.2), a much higher coverage (∼100x) might be necessary to secure a prediction power to retrieve sufficient candidate mutations in such a highly impure sample.

5.3 Conclusions

In this study, we presented a novel probabilistic method for SNV calling, with two significant contributions. First, Virmid can estimate accurate sample composition, or level of contamination of disease sample without genotyping. This not only reduces extra time and cost, but removes severe estimation biases that come from initial SNV calling. Second, Virmid increases genotyping accuracy, especially somatic mutation profiling, by rigorously integrating the sample composition parameter into SNV calling model. We showed Virmid outperformed all recent tools in finding somatic mutations particularly in highly contaminated samples. In applying Virmid to HME disease/normal paired data sets, we discovered previously unknown sample purity and somatic mutations. Our results suggest that it is important to estimate sample composition for all tumor/normal paired data when the sample purity is questionable and explicitly consider the composition in SNV calling if the sample is highly (>50%) contaminated. The robustness of Virmid to high contamination also makes it applicable to mutation identification in other challenging cases, such as low amounts of fetal DNA in maternal plasma.

5.4 Materials and Methods

5.4.1 Virmid Model

Virmid consists of three parts: α estimator, $\mathcal{G}$ estimator, and caller (Figure 5.1). In the α estimator and $\mathcal{G}$ estimator, we use the *Maximum Likelihood Estimation (MLE)* method. The caller calls somatic variants using the Bayesian inference with the estimated joint genotype probability matrix as the prior distribution. To describe the estimation methods in detail, we first define the likelihood function and then describe how the likelihood function is used for each part of Virmid.

Likelihood function

Denote the set of the reads from control sample by C. Given C and a nucleotide position i, the values of the reads mapped to the position i are represented by a vector

(*read vector*) denoted by C^i where the j-th element of C^i (C^i_j) is given by the value (i.e., `A` or `B`) of the j-th mapped read. For the set of the reads from disease-control mixture sample D, D^i is defined similarly.

For the likelihood function, the parameters are α and $\mathcal{G}$. α is the proportion of control in the disease sample. The *joint genotype probability matrix* $\mathcal{G}$ is a 3×3 matrix that describes the average rate of the control-disease joint genotypes g (control genotype) and g' (disease genotype). Given a joint genotype g and g', the element of $\mathcal{G}$ that corresponds to the joint genotype is specified by $\mathcal{G}_{g,g'}$. For example, $\mathcal{G}_{\mathtt{AA},\mathtt{AB}}$ is the rate at which the joint genotype of a random position is `AA` and `AB`. Note that $\mathcal{G}$ is not position-specific; it describes the average rate of each joint genotype over the whole positions. The position specific rates are calculated in the caller using $\mathcal{G}$ as the prior distribution of joint genotypes (see below).

We assume i) all reads at different positions are independent and ii) all reads at the same position are independent given the joint genotype of that position. Given $\theta := \{\alpha, \mathcal{G}\}$, the likelihood function is written by

$$
\begin{aligned}
\mathcal{L}(\theta|C, D) &= P_\theta(C, D) && (5.1)\\
&= \prod_i P_\theta(C^i, D^i) && (5.2)\\
&= \prod_i \Bigg(\sum_{(g,g')\in G} P_\theta(C^i, D^i, g, g') \Bigg) && (5.3)\\
&= \prod_i \Bigg(\sum_{(g,g')\in G} P_\theta(g, g') \cdot P_\theta(C^i, D^i|g, g') \Bigg) && (5.4)\\
&= \prod_i \Bigg(\sum_{(g,g')\in G} P_\theta(g, g') \cdot P_\theta(C^i|g) P_\theta(D^i|g, g') \Bigg) && (5.5)\\
&= \prod_i \Bigg(\sum_{(g,g')\in G} \Bigg\{ \underbrace{P_\theta(g, g')}_{(a)} \cdot \prod_{j=1}^{|C^i|} \underbrace{P_\theta(C^i_j|g)}_{(b)} \cdot \prod_{j=1}^{|D^i|} \underbrace{P_\theta(D^i_j|g, g')}_{(c)} \Bigg\} \Bigg). && (5.6)
\end{aligned}
$$

where G is the set of all possible joint genotypes, and $|C^i|$ and $|D^i|$ denote the read depths of C^i and D^i, respectively. The equations between (5.1) and (5.2), (5.4) and (5.5), and (5.5) and (5.6) are from the above independence assumptions.

The probability in (a) is defined by $P_\theta(g, g') := \mathcal{G}_{g,g'}$. The probabilities in (b) and (c) are defined so that their definitions incorporate the read error rate, mapping

error rate, and the *loss of reads* (LOR) bias. The LOR bias is that reads with more mismatches are less mappable to the reference sequence, which is due to the maximum number of allowed edit distance. First we derive the probabilities in (b) and (c) without considering LOR bias. Denote the read and mapping error probability of C_j^i by r and m, respectively.

The probability in $P_\theta(C_j^i|g)$ is defined by

$$P_\theta(C_j^i|g) := \begin{cases} m\left(\frac{1}{4}\cdot r + (1-r)\beta\right) + (1-m)\left(\frac{1}{4}\cdot r + (1-r)\gamma\right) & \text{if } C_j^i = \texttt{A} \\ 1 - m\left(\frac{1}{4}\cdot r + (1-r)\beta\right) - (1-m)\left(\frac{1}{4}\cdot r + (1-r)\gamma\right) & \text{if } C_j^i = \texttt{B} \end{cases} \tag{5.7}$$

where γ is the probability that an error-free read (i.e., without mapping or read error) has `A` allele given g and β is the probability that an incorrectly mapped read has `A` allele. γ can be calculated as

$$\gamma = \begin{cases} 1 & \text{if } g = \texttt{AA} \\ \frac{1}{2} & \text{if } g = \texttt{AB} \\ 0 & \text{if } g = \texttt{BB}. \end{cases} \tag{5.8}$$

β may depend on experimental settings but is simply set to be $\beta = 0.99$. For simplicity, we denote the right hand side of the equation (5.7) as a function $\mu_g(C_j^i)$.

The probability $P_\theta(D_j^i|g, g')$ in (c) is given by

$$P_\theta(D_j^i|g, g') := \alpha\mu_g(D_j^i) + (1-\alpha)\mu_{g'}(D_j^i). \tag{5.9}$$

The right hand side of the equation (5.9) is denoted by $\nu_g^{g'}(D_j^i, \alpha)$.

Second we derive the probabilities in (b) and (c) considering the LOR bias. Before we derive the probability $P_\theta(C_j^i|g)$ in (b) we first define $x(\texttt{A})$ (or $x(\texttt{B})$), the probability that a read with at least one `A` (or `B`) allele is mappable (i.e., the edit distance of the read is less than the maximum allowed edit distance). Denote the number of mismatches in a read with read length l by #`B`. The distribution of #`B` given one `A` or `B` allele in the read can be derived rigorously per read using the error rates; however, for simplicity, we assume it follows $B(l-1, p)$, the Binomial distribution with $l-1$ trials and the success probability of p. p is the average probability of observing mismatches. We set $p = 0.008$.

Given a maximum number of allowed edit distance d, $x(\mathtt{A})$ is given by

$$\begin{aligned} x(\mathtt{A}) &= P_\theta(\#\mathtt{B} \leq d | \text{an } \mathtt{A} \text{ allele is observed}) \\ &= \sum_{j=0}^{d} \binom{l-1}{j} p^j (1-p)^{l-1-j}. \end{aligned}$$

Likewise, $x(\mathtt{B})$ is given by

$$x(\mathtt{A}) = \sum_{j=0}^{d-1} \binom{l-1}{j} p^j (1-p)^{l-1-j}.$$

To take this probability $x(\cdot)$ into account for $P_\theta(C_j^i|g)$, we think that the read vector (at position i) is generated by two steps: i) generation of a *raw* read vector and ii) generation of the read vector from the raw read vector. The raw read vector is the same as the read vector except that $x(\mathtt{A})$ is assumed to equal $x(\mathtt{B})$ (i.e., the LOR bias is ignored). Then in the second step each element in a raw read vector is retained in the read vector with probability $x(\mathtt{B})$ when the element corresponds to a mismatch or with $x(\mathtt{A})$ otherwise. Thus, for each element in the read vector C_j^i, there is a corresponding element (that C_j^i is from) in the raw read vector. Denote the corresponding element of C_j^i by $\hat{C}_j^i$.

We first derive $P_\theta(\hat{C}_j^i|g)$. Suppose $\hat{C}_j^i = \mathtt{A}$. Define two Bernoulli random variables E_r and E_m as

$$E_r = \begin{cases} 1 & \text{if a read error has occurred for } \hat{C}_j^i \\ 0 & \text{otherwise} \end{cases}$$

and

$$E_m = \begin{cases} 1 & \text{if a mapping error has occurred for } \hat{C}_j^i \\ 0 & \text{otherwise.} \end{cases}$$

We have $P_\theta(E_r = 1) = r$ and $P_\theta(E_r = 1) = m$. The parameter β (the probability that an incorrectly mapped read has $\mathtt{A}$ allele) can be written by $P_\theta(\hat{C}_j^i = \mathtt{A} | E_m = 1, E_r = 0)$, and γ (the probability that an error-free read has $\mathtt{A}$ allele given g) by $P_\theta(\hat{C}_j^i = \mathtt{A} | E_m = 0, E_r = 0, g)$. It is assumed that $\hat{C}_j^i$ and the genotype g are independent when $E_r = 1$ or $E_m = 1$.

Now for $\hat{C}_j^i = \mathtt{A}$, we have

$$P_\theta(\hat{C}_j^i = \mathtt{A}|g) \tag{5.10}$$
$$= P_\theta(\hat{C}_j^i = \mathtt{A}, E_m = 1|g) + P_\theta(\hat{C}_j^i = \mathtt{A}, E_m = 0|g) \tag{5.11}$$
$$= P_\theta(\hat{C}_j^i = \mathtt{A}, E_m = 1) + P_\theta(\hat{C}_j^i = \mathtt{A}, E_m = 0|g) \tag{5.12}$$
$$= P_\theta(E_m = 1) \cdot P_\theta(\hat{C}_j^i = \mathtt{A}|E_m = 1) + P_\theta(E_m = 0|g) \cdot P_\theta(\hat{C}_j^i = \mathtt{A}|E_m = 0, g) \tag{5.13}$$
$$= m \cdot P_\theta(\hat{C}_j^i = \mathtt{A}|E_m = 1) + (1-m) \cdot P_\theta(\hat{C}_j^i = \mathtt{A}|E_m = 0, g) \tag{5.14}$$
$$\begin{aligned} &= m \cdot (r \cdot P_\theta(\hat{C}_j^i = \mathtt{A}|E_m = 1, E_r = 1) + \\ &\quad (1-r) \cdot P_\theta(\hat{C}_j^i = \mathtt{A}|E_m = 1, E_r = 0)) + (1-m) \cdot P_\theta(\hat{C}_j^i = \mathtt{A}|E_m = 0, g) \end{aligned} \tag{5.15}$$
$$= m \cdot (\frac{1}{4} \cdot r + (1-r)\beta) + (1-m) \cdot P_\theta(\hat{C}_j^i = \mathtt{A}|E_m = 0, g) \tag{5.16}$$
$$\begin{aligned} &= m \cdot (\frac{1}{4} \cdot r + (1-r)\beta) + (1-m) \cdot (r \cdot P_\theta(\hat{C}_j^i = \mathtt{A}|E_m = 0, E_r = 1, g) + \\ &\quad (1-r) \cdot P_\theta(\hat{C}_j^i = \mathtt{A}|E_m = 0, E_r = 0, g)) \end{aligned} \tag{5.17}$$
$$= m \cdot (\frac{1}{4} \cdot r + (1-r)\beta) + (1-m) \cdot (\frac{1}{4} \cdot r + (1-r)\gamma). \tag{5.18}$$

If $\hat{C}_j^i = \mathtt{B}$,

$$P_\theta(\hat{C}_j^i = \mathtt{B}|g) = 1 - \big(m \cdot (\frac{1}{4} \cdot r + (1-r)\beta) + (1-m) \cdot (\frac{1}{4} \cdot r + (1-r)\gamma)\big). \tag{5.19}$$

To derive $P_\theta(C_j^i|g)$ from $P_\theta(\hat{C}_j^i|g)$, we define a Bernoulli random variable I_m such that

$$I_m = \begin{cases} 1 & \text{if } \hat{C}_j^i \text{ is retained in the read vector} \\ 0 & \text{otherwise .} \end{cases}$$

We have $P_\theta(I_m = 1|\hat{C}_j^i = \mathtt{A}) = x(\mathtt{A})$ and $P_\theta(I_m = 1|\hat{C}_j^i = \mathtt{B}) = x(\mathtt{B})$. Since C_j^i we observe is always retained from $\hat{C}_j^i$ (i.e., $I_m = 1$ is given), the probability $P_\theta(C_j^i|g)$ can be rewritten by $P_\theta(\hat{C}_j^i|g, I_m = 1)$, which is proportional to $P_\theta(\hat{C}_j^i|g) \cdot P_\theta(I_m = 1|\hat{C}_j^i)$. Therefore, if we denote $P_\theta(\hat{C}_j^i|g)$ by $\mu_g(\hat{C}_j^i)$, we obtain

$$P_\theta(C_j^i|g) = \frac{\mu_g(C_j^i) \cdot x(C_j^i)}{\mu_g(\mathtt{A}) \cdot x(\mathtt{A}) + \mu_g(\mathtt{B}) \cdot x(\mathtt{B})} \tag{5.20}$$

The right hand side of the equation (5.20) is denoted by $f_g(C_j^i)$.

Next we derive the probability $P_\theta(D_j^i|g,g')$ as follows: Denote the corresponding element of D_j^i in the raw read vector by $\hat{D}_j^i$ as above. Define a Bernoulli random variable I_C by

$$I_C = \begin{cases} 1 & \text{if } \hat{D}_j^i \text{ is from the control sample} \\ 0 & \text{otherwise .} \end{cases}$$

The success probability of I_C is given by α. We have

$$\begin{aligned} P_\theta(\hat{D_j^i}|g,g') &= P_\theta(\hat{D_j^i}, I_C = 0|g,g') + P_\theta(\hat{D_j^i}, I_C = 1|g,g') && (5.21)\\ &= P_\theta(\hat{D_j^i}, I_C = 0|g') + P_\theta(\hat{D_j^i}, I_C = 1|g) && (5.22)\\ &= (1-\alpha)P_\theta(\hat{D_j^i}|I_C = 0,g') + \alpha P_\theta(\hat{D_j^i}|I_C = 1,g) && (5.23)\\ &= (1-\alpha)\mu_{g'}(\hat{D_j^i}) + \alpha\mu_g(\hat{D_j^i}). && (5.24) \end{aligned}$$

Denote $P_\theta(\hat{D_j^i}|g,g')$ by $\nu_g^{g'}(\hat{D_j^i},\alpha)$. As in (5.20), we obtain

$$P_\theta(D_j^i|g,g') = \frac{\nu_g^{g'}(D_j^i,\alpha)\cdot x(D_j^i)}{\nu_g^{g'}(\mathtt{A},\alpha)\cdot x(\mathtt{A}) + \nu_g^{g'}(\mathtt{B},\alpha)\cdot x(\mathtt{B})} \tag{5.25}$$

The right hand side of the equation (5.25) is denoted by $h_g^{g'}(D_j^i,\alpha)$.

Overall, we have

$$\mathcal{L}(\theta|C,D) = \prod_i \Bigg(\sum_{(g,g')\in G} \Big\{ \mathcal{G}_{g,g'} \cdot \prod_{j=1}^{|C^i|} f_g(C_j^i) \cdot \prod_{j=1}^{|D^i|} h_g^{g'}(D_j^i,\alpha) \Big\} \Bigg). \tag{5.26}$$

Basic model

Using the likelihood function defined above, the MLE of $\theta = (\alpha,\mathcal{G})$ can be obtained by

$$\hat{\theta} = \arg\max_\theta \mathcal{L}(\theta|C,D) \tag{5.27}$$

with proper constraints. The constraints are different in the α estimation step and $\mathcal{G}$ estimation step. This is because in the α estimation step we are only estimating α and a subset of $\mathcal{G}$. Moreover, some of elements of $\mathcal{G}$ are expected to have different frequencies in the α estimation step than in the $\mathcal{G}$ estimation step. We remark that the constraints we used are relatively liberal; they were used just to reduce the search space.

Constraints for α estimation

$$
\begin{aligned}
0 \leq & \alpha \leq 1 \\
0 \leq & \mathcal{G}_{\mathtt{AA,AA}} \leq 1 \\
0 \leq & \mathcal{G}_{\mathtt{AA,AB}} \leq 1 \\
0 \leq & \mathcal{G}_{\mathtt{AA,BB}} \leq 1 \\
10 \cdot & \mathcal{G}_{\mathtt{AA,BB}} \leq \mathcal{G}_{\mathtt{AA,AB}} \\
\mathcal{G}_{\mathtt{AA,AA}} + & \mathcal{G}_{\mathtt{AA,AB}} + \mathcal{G}_{\mathtt{AA,BB}} = 1
\end{aligned}
$$

Constraints for $\mathcal{G}$ estimation

$$
\begin{aligned}
0 \leq & \alpha \leq 1 \\
0 \leq & \mathcal{G}_{g,g'} \leq 1 \text{ for } g, g' \in \{\mathtt{AA,AB,BB}\} \\
10^2 \cdot & (\mathcal{G}_{\mathtt{AB,AA}} + \mathcal{G}_{\mathtt{AB,AB}} + \mathcal{G}_{\mathtt{AB,BB}}) \leq (\mathcal{G}_{\mathtt{AA,AA}} + \mathcal{G}_{\mathtt{AA,AB}} + \mathcal{G}_{\mathtt{AA,BB}}) \\
10^2 \cdot & (\mathcal{G}_{\mathtt{BB,AA}} + \mathcal{G}_{\mathtt{BB,AB}} + \mathcal{G}_{\mathtt{BB,BB}}) \leq (\mathcal{G}_{\mathtt{AB,AA}} + \mathcal{G}_{\mathtt{AB,AB}} + \mathcal{G}_{\mathtt{AB,BB}}) \\
10^4 \cdot & \mathcal{G}_{\mathtt{AA,AB}} \leq \mathcal{G}_{\mathtt{AA,AA}} \\
10^2 \cdot & \mathcal{G}_{\mathtt{AA,BB}} \leq \mathcal{G}_{\mathtt{AA,AB}} \\
10^6 \cdot & \mathcal{G}_{\mathtt{AB,AA}} \leq \mathcal{G}_{\mathtt{AB,AB}} \\
10^2 \cdot & \mathcal{G}_{\mathtt{AB,BB}} \leq \mathcal{G}_{\mathtt{AB,AB}} \\
10^6 \cdot & \mathcal{G}_{\mathtt{BB,AB}} \leq \mathcal{G}_{\mathtt{BB,BB}} \\
10^6 \cdot & \mathcal{G}_{\mathtt{BB,AA}} \leq \mathcal{G}_{\mathtt{BB,BB}} \\
\sum_{g,g' \in \{\mathtt{AA,AB,BB}\}} & \mathcal{G}_{g,g'} = 1
\end{aligned}
$$

Since the exact global maximum point cannot be derived analytically, one needs to use a numerical approach to find it. To make a numerical approach work, one should carefully avoid the local maximum points. However, even if we impose strong constraints, many local maximum points may be present in the likelihood function. Moreover, in terms of the estimation of α, not all read vectors are useful; some read vectors deteriorate the estimation (see below). Therefore, we try to estimate α and then estimate all elements in $\mathcal{G}$ (with the estimated α).

Estimation of α

For the estimation of α, the disease read vectors generated from the same control genotype (g) and disease genotype (g') are simply noisy sample points conveying no information. Thus, we want to sample the read vectors generated from different g and g', but without the initial calling. Also we want to fix $g =$`AA` so that the number of parameters to be estimated can be minimized. Denote the number of `B`'s in a control read vector C^i (or in a disease read vector D^i) by $\langle C^i \rangle$ (or $\langle D^i \rangle$). We sample the positions i such that $\langle C^i \rangle = 0$ and $\frac{\langle D^i \rangle}{|D^i|} > R$ (i.e., the BAF of D^i is larger than R) for a real value $0 < R \leq 1$. Imposing $\langle C^i \rangle = 0$ minimizes the chance that $g =$`AA`, and imposing $\frac{\langle D^i \rangle}{|D^i|} > R$ increases the chance that $g' \neq$`AA`.

If R is too large, however, we may not have sufficient number of samples for the estimation. On the other hand, if R is too small, the samples may contain too many read vectors from $g =$`AA` and $g' =$`AA` that serve as noise. Thus, we estimate α using different values of R and take the median of the estimates. Table 5.1 shows that our α estimator is quite robust for different values of R. We also outputs the asymptotic variance of the estimated α using the outer products of the first derivatives of the log likelihoods (called *BHHH* estimator [BHH74]).

With the selected samples as above, we only estimate 4 parameters (instead of 10 - α and 9 elements in $\mathcal{G}$): α, $\mathcal{G}_{\texttt{AA,AA}}$, $\mathcal{G}_{\texttt{AA,AB}}$, and $\mathcal{G}_{\texttt{AA,BB}}$; other elements in $\mathcal{G}$ are set to a very small number close to 0. In this step, the parameters except α are estimated simply for better estimation of α.

Unfortunately, the sampling described above introduces estimation bias if we use the likelihood function as is because the sampling procedure inflates the number of `B`'s in the disease read vector. To take this sampling bias into account, we modify the likelihood function as

$$\mathcal{L}_R(\theta|C, D) = P_\theta(C, D|\frac{\langle D^i \rangle}{|D^i|} > R \text{ for all } i) \tag{5.28}$$

$$= \prod_i \Big(\sum_{(g,g') \in G} \Big\{ \mathcal{G}_{g,g'} \cdot \prod_{j=1}^{|C^i|} f_g(C^i_j) \cdot \underbrace{\frac{\prod_{j=1}^{|D^i|} h_g^{g'}(D^i_j, \alpha)}{P_\theta\Big(\frac{\langle D^i \rangle}{|D^i|} > R|g, g'\Big)}}_{(d)} \Big\} \Big). \tag{5.29}$$

The denominator in (d) can be efficiently calculated using a dynamic programming algorithm with the time complexity of $O(R \cdot |D^i|^2)$ as follows: Denote $R \cdot |D^i|$ by N. Then, $P_\theta\Big(\frac{\langle D^i \rangle}{|D^i|} > R|g, g'\Big) = P_\theta\Big(\langle D^i \rangle > N|g, g'\Big) = 1 - P_\theta\Big(\langle D^i \rangle \leq N|g, g'\Big)$. We calculate $P_\theta\Big(\langle D^i \rangle \leq N|g, g'\Big)$ using a dynamic programming.

Let $H(j, n)$ be the probability that the n elements among $D^i_1, \cdots, D^i_j$ are $\mathtt{B}$ alleles. Then, when $j > 1$ and $n > 0$ we have

$$H(j, n) = H(j-1, n) \cdot P_\theta(D^i_j = \mathtt{A}|g, g') + H(j-1, n-1) \cdot P_\theta(D^i_j = \mathtt{B}|g, g'). \quad (5.30)$$

Since $H(j, 0) = H(j-1, 0) \cdot P_\theta(D^i_j = \mathtt{A}|g, g')$ for $n = 0$ and $P_\theta(D^i_j|g, g') = h^{g'}_g(D^i_j)$, we obtain the following recursion:

$$H(j, n) = H(j-1, n) \cdot h^{g'}_g(D^i_j = \mathtt{A}, \alpha) + H(j-1, n-1) \cdot h^{g'}_g(D^i_j = \mathtt{B}, \alpha) \quad (5.31)$$

for $n \geq 0$ and $j \geq 1$. The boundary conditions are given by $H(0, 0) := 1$ and $H(j, -1) := 0$. The probability $P_\theta\Big(\frac{\langle D^i \rangle}{|D^i|} > R|g, g'\Big) = 1 - P_\theta\Big(\langle D^i \rangle \leq N|g, g'\Big)$ is calculated by $1 - \sum_{n=0}^{N} H(|D^i|, n)$. The time complexity is given by $O(|D^i| * N) = O(|D^i|^2 \cdot R)$.

As above, the parameter R can be readily incorporated in our model in our method (correcting the possible bias); however, it is very hard to make a rigorous model that takes the LOV (loss of variants) bias found in the calling-based methods into account.

The estimates of the 4 parameters ($\hat{\alpha}$, $\hat{\mathcal{G}}_{\mathtt{AA},\mathtt{AA}}$, $\hat{\mathcal{G}}_{\mathtt{AA},\mathtt{AB}}$, and $\hat{\mathcal{G}}_{\mathtt{AA},\mathtt{BB}}$) that maximize the likelihood are found by the *feasible direction method*[BSS05], a gradient descent search method with constraints. Note that only the estimate of α is retained for the next step.

Figure 5.4 A shows the values of log likelihood over different $\hat{\alpha}$ and $\hat{\mathcal{G}}_{\mathtt{AA},\mathtt{AB}}$ (for each point, other parameters are optimized). For low α, the optimum $\hat{\mathcal{G}}_{\mathtt{AA},\mathtt{AB}}$ is almost 1. However, when α is larger, the likelihood is maximized for low $\hat{\mathcal{G}}_{\mathtt{AA},\mathtt{AB}}$. For example, when α= 0.9, the maximum likelihood is found when $\hat{\mathcal{G}}_{\mathtt{AA},\mathtt{AB}} \approx 0.1$. Such estimation results are predicted because for high α, even disease read vectors generated with g' =AB would not have a sufficient number of B's to distinguish between g' =AB and g' =AA. Even if we sample disease read vectors with many B's, there are often many vectors from g' =AA which leads to high value of $\hat{\mathcal{G}}_{\mathtt{AA},\mathtt{AB}}$.

Estimation of the joint genotype probability matrix

In this step, we estimate $\mathcal{G}$ with the estimated α . We sample 1,000,000 positions except ones at which the number of `B` in the disease read vector is zero (i.e., $\langle D^i \rangle = 0$). Such positions are not sampled because the Virmid does not analyze such points for SNP calling. We estimate $\hat{\mathcal{G}}$ that maximizes this likelihood function in (5.26) using the feasible direction method.

Calling genotypes

Given the estimated $\hat{\theta} = (\hat{\alpha}, \hat{\mathcal{G}})$ and a position i, we first calculate $\mathcal{G}^i$, the posterior distribution of genotypes at the position i, (with $\hat{\mathcal{G}}$ as the prior distribution) by

$$\mathcal{G}^i_{g,g'} = P_{\hat{\theta}}(g, g' | C^i, D^i) \tag{5.32}$$

$$= \frac{P_{\hat{\theta}}(C^i, D^i | g, g') \cdot P_{\hat{\theta}}(g, g')}{\sum_{(k,k') \in G} P_{\hat{\theta}}(C^i, D^i | k, k') \cdot P_{\hat{\theta}}(k, k')} \tag{5.33}$$

$$= \frac{\hat{\mathcal{G}}_{g,g'} \cdot \prod_{j=1}^{|C^i|} f_g(C^i_j) \cdot \prod_{j=1}^{|D^i|} h^{g'}_g(D^i_j, \hat{\alpha})}{\sum_{(k,k') \in G} \hat{\mathcal{G}}_{k,k'} \cdot \prod_{j=1}^{|C^i|} f_k(C^i_j) \cdot \prod_{j=1}^{|D^i|} h^{k'}_k(D^i_j, \hat{\alpha})}. \tag{5.34}$$

Then Virmid calls the position i a somatic variant if $1 - (\mathcal{G}^i_{\mathtt{AA},\mathtt{AA}} + \mathcal{G}^i_{\mathtt{AB},\mathtt{AB}} + \mathcal{G}^i_{\mathtt{BB},\mathtt{BB}})$ exceeds 0.5.

Filtration of read data

Reads or positions that may contain unreliable information were filtered out from observation. Two filtering criteria have been established depending on applying Virmid step. The first filtering scheme is for selecting observation points for α estimation. The purpose of filtering in this step is to eliminate positions possibly contain following noises: 1) `B` alleles originated from sequencing error, 2) `B` alleles originated from mapping error, 3) `B` alleles originated from non-reference control genotype. The second filtering scheme is for calling somatic mutations. The purpose in this step is to remove false-positive somatic mutations, which are usually one of the following

cases: 1) both samples have reference genotype (`AA-AA`), but `B` alleles are observed by sequencing or mapping errors, 2) both samples have non-reference genotype (`AB-AB`, germline mutation), but significant BAF differences are observed. To satisfy above criteria, we divided all frequent miscalling events into seven classes. 1) mapping quality (MQ): mutations are filtered out if the mapping quality of their corresponding `B` allele read is significantly worse ($>$ 30 MAPQ score) than `A` allele reads. Or overall ratio of ambiguously mapped reads ($<$ 17 MAPQ score) is more than a threshold ($>$ 0.4). 2) read offset filter: if the position of `B` allele in the read is significantly biased at the both ends. (z-score$>$ 3). 3) indel proximity (PRX): if more than 50% of `B` alleles are located within 10 bp of nearby indels. 4) tri-allele (TRI): if the major allele frequency is less than 0.9. 5) base quality (BQ): if the mean base call quality of `B` allele read is less than 20. 6) number of mismathc (NM): the mean number of mismatches per read is bigger than 3. Or more than 60% of the reads are soft/hard clipped. 7) allele frequency (AF): if the absolute number of `B` allele is less than a certain threshold (3), or BAF in control is larger than one tenth (1/10) of that in disease sample. The filters are differently applied in α estimation and mutation calling. For α estimation, our goal is to eliminate germline mutations (`AB-AB`) only allowing reference and somatic mutation alleles. To do so, we strictly apply MQ, PRX and NM filters to prevent potential mapping errors. For mutation calling, we apply all the seven filters with empirically known parameters. These parameters can be also defined by users.

5.4.2 Data preparation

Simulated data

First, two diploid genomes were simulated: a normal genome and a disease genome. The normal genome was created by using the hg19 genome as a template and infusing germline SNPs found in dbSNP 135[SWK$^+$01] at a rate of one SNP per thousand nucleotides. Somatic mutations were introduced by perturbing a nucleotide to any of the other three nucleotides with equal probability at rates of 10^{-5} mutations per nucleotide to simulate a disease genome. Both of these simulations were carried out using in-built python functions. The python scripts are available upon request.

GemSim v1.5[MLT12] was setup to generate paired-end 100 bp reads using the Illumina paired-end error model. The number of reads necessary was calculated using the average coverage of the sample (40x and 100x). The *metagenomic* mode was configured with four genomes: normal haplotype 1, normal haplotype 2, disease haplotype 1, and disease haplotype 2. The relative abundance of each genome was calculated based on the contamination level (α=1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90%). For the normal sample, the metagenomic mode was configured with the normal haplotype 1 and normal haplotype 2 in equal abundances. All the reads were aligned using bwa[LD10] and passed through the GATK data processing pipeline for variant calling including indel realignment and base quality score recalibration. The resulting BAM files were fed into the variant calling tools.

Breast cancer data

There were 545 tumor to normal matched samples with verified somatic mutations. The putative somatic mutations were validated using Illumina Capture gDNA technologies. For our purposes of assessing normal cell contamination in tumor specimens, we downloaded the matched tumor/normal samples listed as whole exome sequencing (WXS) on CGHub (https://cghub.ucsc.edu/) under controlled access. As essential post-processing on the mapping such as indel realignment and quality recalibration is time-consuming, we limit our analysis to randomly selected 15 patients for use as our gold standard. (see Table 5.3). The 1654 validated mutations were extracted from accompanying mutation annotation format (MAF) file by the following criteria: 1) field "Validation_Status" (column 25) is "Valid", 2) field "Variant_Type" (column 10) is "SNP".

HME data

Five paired normal data sets (10 BAM files, 76 bp, 30x coverage) processed in the previous study [LHS$^+$12] were downloaded with authors' permission. The alignments are already post-processed using GATK's pipeline including IndelRealigner, MarkDuplicate and TableRecalibration. Pileup files are generated using samtools *mpileup* and indexed with tabix. Possible noise reads that do not pass the quality

check or possibly included as PCR duplicates were filtered out using samtools view -F option.

Call-based estimation of α

Initial SNV calling was done using Virmid w/o α mode. All the filtration steps are applied after the initial calling. The detailed calculation steps are introduced in the supporting information of Snyder *et al* [SKVQ11]. Briefly, the number of reads at the called somatic mutation sites are classified by genotypes and allele types. Donor fraction is estimated from $\frac{2N_{AB}(B)+N_{BB}(B)}{N_{AB}(A)+N_{AB}(B)+N_{BB}(A)+N_{BB}(B)}$ where $N_G(A)$ denotes the number of reads at the site of genotype G with A allele.

Somatic mutation call sets

Strelka version 0.4.5 was used for the comparative studies. The program was configured using the provided settings for bwa. The results presented show the calls after the first filtration step and after the final filtration step. For JointSNVMix version 0.7[RMD+12], the results were generated using the JointSnvMix2 mode, which included the base and mapping errors. First the program was trained using the jsm.py train, and then joint genotype calls were made using the jsm.py classify option. All of the configurable settings were left to their default values. For the AUC curves, we varied the probability cutoff necessary to make a joint genotype call to adjust the specificity and sensitivity of the program. For the filtered version of JointSNVMix2, we used JointSNVMix version 0.8 with "–post_process option". Other parameters were the same as JointSNVMix version 0.7. For VarScan2[KZL+12], we evaluated its performance using version 2.2.11. The pileup files were created using samtools version 0.1.18 [LHW+09]. The somatic option was adjusted with the Virmid derived values of contamination for tumor purity calculations. Note that since VarScan2 requires purity estimation, the values fed into the program were (1-predicted contamination level). We also carried out additional filtering using the default options for VarScan2's methods somaticFilter and processFilter as well as the perl script fpfilter.pl that is available on VarScan2's website. The most recent version of bam-readcount per its github repository was used to create the input files for the perl script. Lastly, we ran

MuTect as described in the MuTect website,except for "–cosmic" option since the validated mutations (true answer) are included in the corresponding database.

5.4.3 Program implementation and optimization

We implemented the Virmid model and its surrounding workflow using Java (JDK version 1.6), samtools and picard library. Post-processed BAM files are converted to pileup format using samtools *mpileup* program. Mapping quality scores are included in the pileup files using '-s' option. To optimize the overall pipeline, we divided the pileup data into three different layers (Figure 5.2 A). Pileup level 1 is the most fundamental data in which we observe `B` allele at least one time in the disease genome. Level 2 data contains all the positions that the observed BAF is higher than or equal to 5% as well as there is no `B` allele observed in the control genome. Lastly, level 3 data was generated by increasing the minimum BAF until the number of satisfying positions is less than a threshold (generally 1k to 10k). α estimation is done using the level 3 disease pileup data. After getting α, we call genotypes of the positions included in the level 1 pileup files. This hierarchical model significantly reduced overall serach space (Figure 5.2 B). Starting from all the nucleotide regions of chromosome 1 ($\sim$240 Mbp), the size of level 1 data is reduced to 9%. Final numbers of data points are reduced to 0.28% and 0.00041% of the original number in level 2 and level 3 data respectively. Due to the successful reduction, we decreased the running time for α estimation to less than a few minutes.

Acknowledgements

Chapter 5, in full, was submitted as "Virmid: virtual microdissection of sample mixtures for accurate somatic mutation profiling". S. Kim, K. Jeong, K. Bhutani1, J. Lee, A. Patel, E. Scott, H. Nam, H. Lee, J. G. Gleeson, and V. Bafna. Sangwoo Kim and the dissertation author were the primary authors of this paper.

Table 5.1: Virmid was ran 20 times for each sample to estimate α with variance. Generally, all the estimation was clustered very close to the true value. [a] Estimates without the *loss of read* (LOR) bias correction. The probability that a read with at least one `A` is mappable was set to equal the probability that a read with at least one `B` is mappable, that is, $x(\texttt{A}) = x(\texttt{B})$ (see Methods). Note that *loss of variants* (LOV) bias was already addressed here. [b] Range of minimum BAF in the observed data. The more data points Virmid observes, the smaller minimum BAF is obtained, which is used in Virmid model to correct sampling bias.

True α(%)	Estimated. α(%)			Stdev (10^{-2})	α range	BAF range [b]
	Call-based	Virmid$_{(/\text{LOR})}$[a]	Virmid			
1	2.64	2.56	**1.61**	0.19	1.23∼1.90	11.11∼47.50
5	6.31	5.56	**4.74**	0.28	4.07∼5.17	11.11∼46.34
10	10.3	10.50	**9.86**	0.51	9.34∼11.03	11.43∼44.12
20	20.4	19.92	**19.59**	0.44	18.44∼20.13	11.43∼38.46
30	30.4	30.33	**30.28**	0.48	28.48∼30.79	11.11∼33.33
40	39.6	40.46	**39.94**	0.22	39.49∼40.51	11.43∼28.89
50	49.2	51.38	**50.72**	0.23	50.46∼51.31	11.11∼23.53
60	56.2	61.16	**60.62**	0.54	59.78∼61.48	10.71∼19.15
70	62.4	70.28	**70.05**	0.16	69.82∼70.44	10.00∼14.29
80	67.2	80.33	**80.04**	0.38	79.52∼80.76	9.38∼10.81
90	67.4	88.91	**88.88**	0.53	88.06∼90.00	8.82∼9.68

Table 5.2: Precision and recall values are compared between two different coverages (40x and 100x) at highly contaminated data (α=70%, 80%, and 90%). Much deeper coverage is required when severe contamination is expected. α=proportion of control sample in disease sample (contamination level), n.Answer=number of true somatic mutations, n.Predict=number of predicted mutations by Virmid, n.Correct=number of correct mutations from prediction, Prc=Precision, Rec=Recall, [a]rounded up from 0.9954, [b]rounded up from 0.9958

α	n.Answer	Coverage=40x					Coverage=100x				
		n.Predict	n.Correct	Prc	Rec	F-score	n.Predict	n.Correct	Prc	Rec	F-score
70%	2265	1999	1976	0.98	0.87	0.93	2208	2198	1.00[a]	0.97	0.98
80%		1551	1516	0.98	0.67	0.79	2142	2133	1.00[b]	0.94	0.97
90%		638	611	0.96	0.27	0.42	1572	1545	0.98	0.68	0.81

Table 5.3: A list of 15 public breast cancer data from TCGA.

Tumor sample ID	Normal sample ID (matched)
TCGA-BH-A0DZ-01A-11W-A019-09	TCGA-BH-A0DZ-10A-01W-A021-09
TCGA-A2-A0T0-01A-22D-A099-09	TCGA-A2-A0T0-10A-01D-A099-09
TCGA-A1-A0SO-01A-22D-A099-09	TCGA-A1-A0SO-10A-03D-A099-09
TCGA-A8-A06X-01A-21W-A019-09	TCGA-A8-A06X-10A-01W-A021-09
TCGA-A8-A093-01A-11W-A019-09	TCGA-A8-A093-10A-01W-A021-09
TCGA-AO-A0J9-01A-11W-A050-09	TCGA-AO-A0J9-10A-01W-A055-09
TCGA-AN-A0FJ-01A-11W-A019-09	TCGA-AN-A0FJ-10A-01W-A021-09
TCGA-A8-A095-01A-11W-A019-09	TCGA-A8-A095-10A-01W-A021-09
TCGA-BH-A0DK-01A-21W-A071-09	TCGA-BH-A0DK-10A-01W-A071-09
TCGA-A7-A0CE-01A-11W-A019-09	TCGA-A7-A0CE-10A-01W-A021-09
TCGA-AR-A0TV-01A-21D-A099-09	TCGA-AR-A0TV-10A-01D-A099-09
TCGA-A8-A0A7-01A-11W-A019-09	TCGA-A8-A0A7-10A-01W-A021-09
TCGA-BH-A0C0-01A-21W-A071-09	TCGA-BH-A0C0-10A-01W-A071-09
TCGA-AO-A0JL-01A-11W-A071-09	TCGA-AO-A0JL-10A-01W-A071-09
TCGA-A2-A0CT-01A-31W-A071-09	TCGA-A2-A0CT-10A-01W-A071-09
TCGA-A8-A07L-01A-11W-A019-09	TCGA-A8-A07L-10A-01W-A021-09

Table 5.4: Three previously validated mutations have deviated mutation burdens compared to the expected values (50%). The mutation burdens were measured using mass spectrometry (MS) or whole exome sequencing (WES) from previous study. Samples are collected from different sites of the affected brain hemisphere. CO=Central operculum, Pa=Parietal, Te=Temporal, Or=Orbital, Fr=Frontal and Oc=Occipital

Subject	Mutation	Method	Mutation burden (Sampling lesion)	Average (%)
HME-1563	MTOR c.4448C>T	MS	30.3%, 8.3%, 36.4%, 9.1%, 18.1% (CO)	20.44%
		WES	9.7% (17/176)	9.7%
HME-1565	AKT3 c.49C>T	MS	26.9% (Pa), 30.4% (CO), 15.6% (Te)	24.3%
		WES	28.0% (9/32)	28.0%
HME-1573	PIK3CA c.1633G>A	MS	25.0% (Or), 36.0% (Fr), 39.5% (CO), 21.0% (Oc)	30.38%
		WES	16.0% (9/56)	16.0%

Table 5.5: List of predicted somatic mutations in five HME individual samples are shown. Predicted mutations sets are compared to previous study [LHS$^+$12] using JointSNVMix2. Virmid called more novel (and less overlapping) mutations in samples with higher α (HME-1574 and HME-1620). Est.α=Estimated contamination level (α), std.=standard deviation, n.Total.Mut=number of all called mutations, n.Mut=number of (overlapping or novel) mutations, n.Exon=number of mutations in exon, n.Mis=number of missense (or nonsense) mutations. *one mutation loci may be counted multiple times when more than one genetic products exist at the position.

			Overlapping mutation				Novel mutation			
Subject	Est.α (std.)	n.Total.Mut	n.Mut*	n.Exon	n.Mis	Gene	n.Mut*	n.Exon	n.Mis	Gene
HME-1563	77.9% ($\pm$0.006)	478	108	54	33	MTOR	370	209	147	
HME-1565	71.3% ($\pm$0.003)	494	124	78	50	AKT3	370	72	43	
HME-1573	64.0% ($\pm$0.017)	542	235	167	112	PIK3CA	307	301	185	
HME-1574	83.4% ($\pm$0.005)	579	100	63	34		479	412	281	MTOR
HME-1620	84.8% ($\pm$0.004)	694	86	56	27		608	383	267	

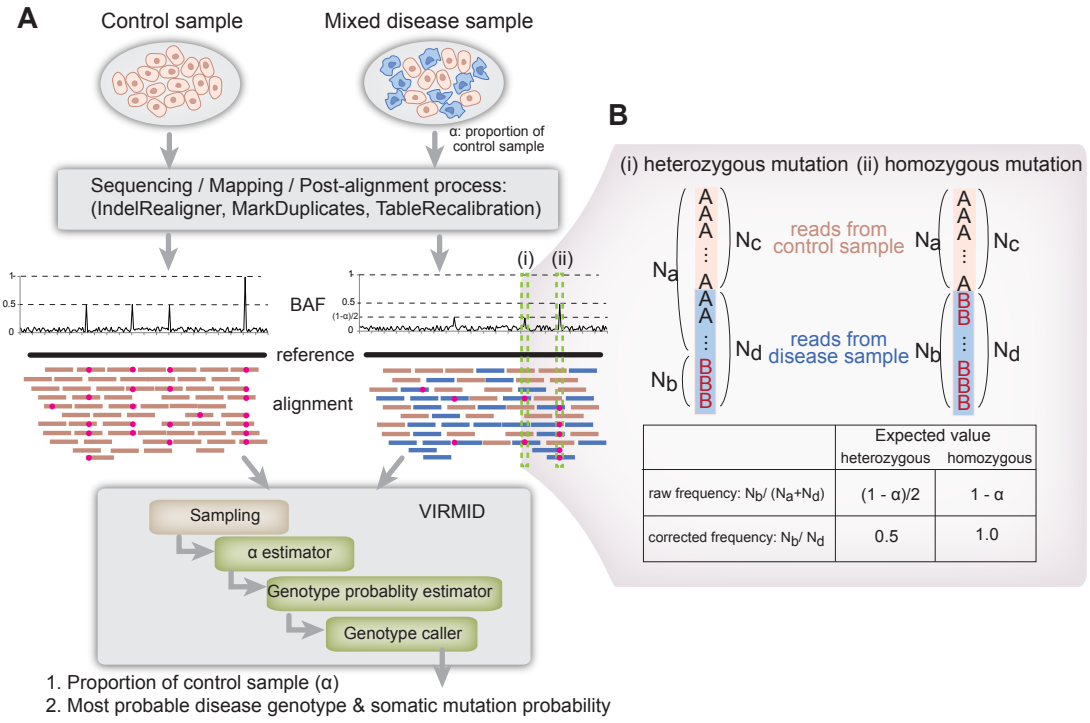


Figure 5.1: The complete Virmid workflow is shown. (a) Disease/control paired data are given (top) to generate an alignment (BAM) file. The mixed disease sample produce short reads of mixed types (blue and orange rectangles). Somatic mutations, where the control has the reference genotype (`AA`) and the disease has non-reference (`AB` or `BB`, red dots in alignment), are hard to detect in high contamination due to the significant drop in B-allele frequency (BAF). Virmid takes the disease/control paired data and analyzes 1) the sample level proportion of control cells in the disease sample (α) and 2) the most probable disease genotype for each position that can be used to call somatic mutations. (b) An example of BAF drop is shown. Without contamination, the expected BAF is 0.5 and 1.0 for heterozygous and homozygous mutations sites respectively. When there is a control sample contamination of α, mutation alleles are observed only in (1-α) of the whole reads. So the expected BAF is dropped to $(1-\alpha)/2$ and $(1-\alpha)$. With estimating the accurate α, Virmid can detect more true somatic mutations that can be missed in the conventional tools due to the insufficient observation of `B` alleles

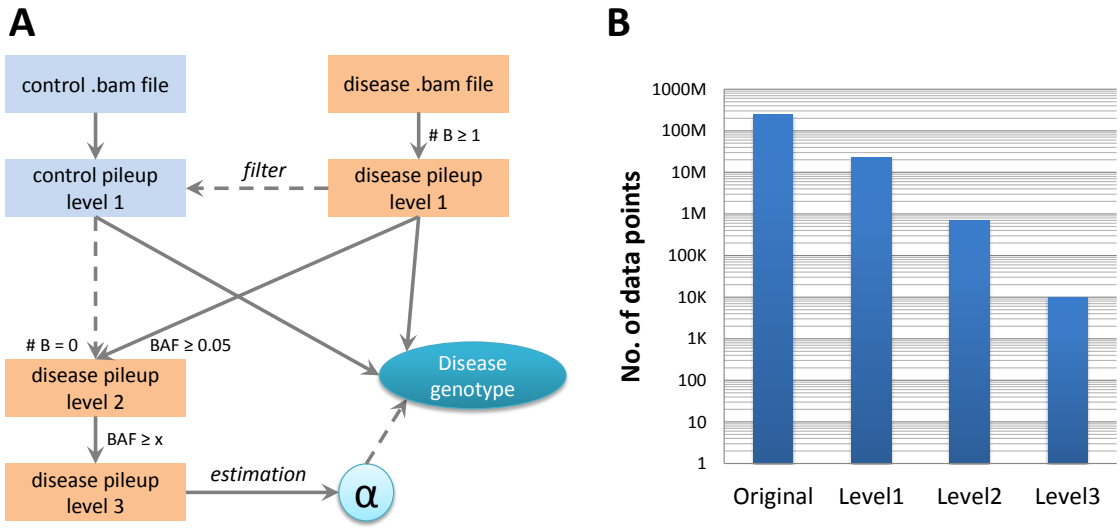


Figure 5.2: Multi-tier sampling of Virmid. A, given disease BAM file is first reduced to a smaller subset in which at least one `B` allele is observed. In control BAM sample, only positions in which no `B` allele is observed are used. These samples are further filtered out using minimum BAF to increase the probability of selecting true somatic mutations. B, the size of sample is dramatically reduced down to 0.01% of the initial data.

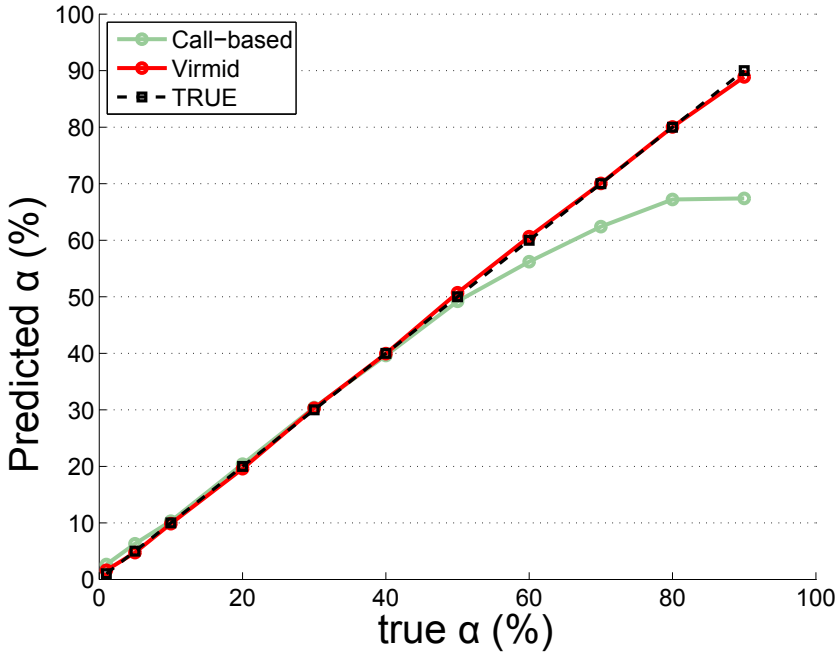


Figure 5.3: Estimation of contamination level in mixed disease sample. The proportion of control sample (α) is estimated from the simulated mixed data. A total of 11 data sets with different α(1%, 5%, 10% 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%) were generated and tested. Virmid estimated all the α (red line with circles) with high concordance compared to the true values (black line with squares). Note that there is a significant bias in highly contaminated samples ($\alpha \geq 60$) in the call based method (green line with circles) due to undetectable low BAF mutations; somatic mutations with higher BAF are likely to be called initially causing overestimation of BAF and underestimation of α.

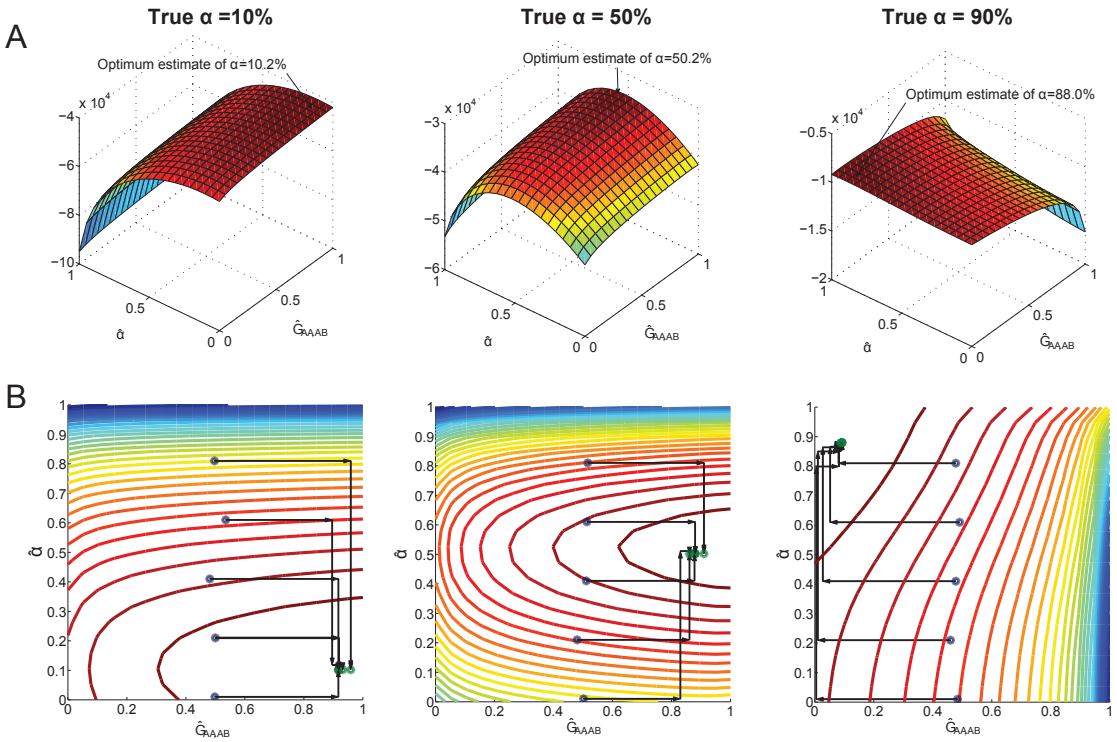


Figure 5.4: (a) Log likelihoods over different $\hat{\alpha}$ (the estimate of α) and $\hat{\mathcal{G}}_{\mathtt{AA,AB}}$ (the estimate of $\mathcal{G}_{\mathtt{AA,BB}}$, the probability that the control genotype is `AA` and the disease genotype is `AB`). The surface graph shows that likelihoods are maximized around the true α values (10%(left), 50%(middle), 90%(right)). (b) Searching paths using the feasible direction method is denoted on contour maps. The method efficiently finds the optimum points (green circles) only in a few searching steps. Searching processes from different starting points (blue circles) are finally converged.

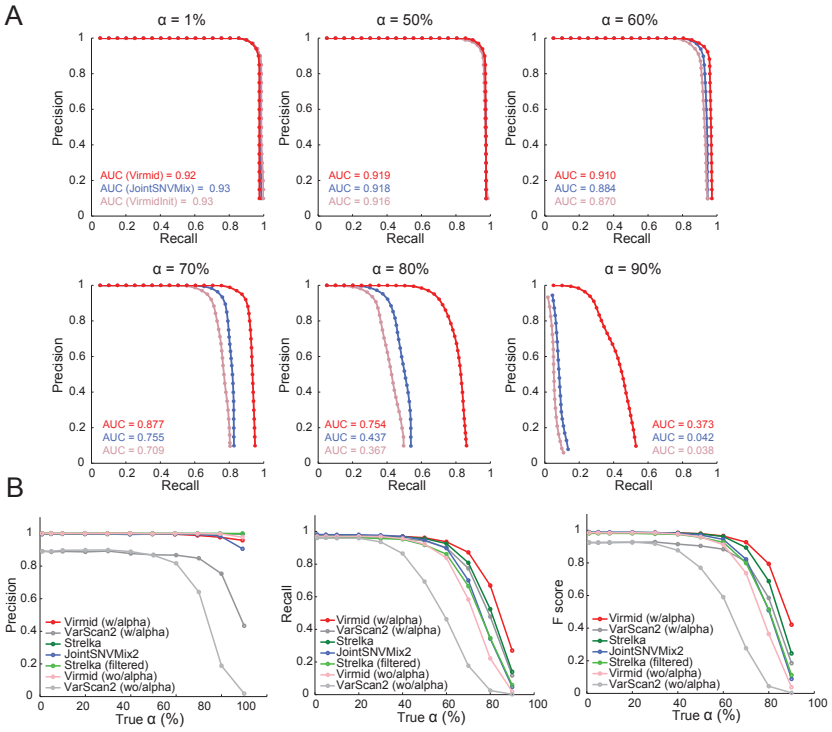


Figure 5.5: Performance comparison of different methods for somatic mutation detection. (a) Precision-recall curves of Virmid with α (red), Virmid without α (light red), and JointSNVMix2 (blue) at six different α (1%, 50%, 60%, 70%, 80% and 90%) are shown. Note that the performance is significantly improved when α is incorporated into the calling model. There is little difference in performance at low contamination levels ($\alpha \leq 50$). (b) Precision and recall scores of final call are generated for each α, where mutation probabilities are not available; note that a single point instead of a curve is plotted for each α. As α grows, there is a consistent drop in precision, recall, and f-score*. Four tools including Virmid, Strelka, VarScan2 and JointSNVMix2 are tested with the same data. Virmid and VarScan2 are tested in two different modes (with and without α). Strelka was also tested in two modes with or without applying quality control. Overall, Virmid with α scored best in f-score, followed by Strelka and VarScan2 with α and JointSNVMix2. Note that the tools with α (Virmid with α, Strelka, VarScan2 with α) outperformed those without α (Virmid without $\alpha$and VarScan2 without α), denoting that importance of incorporating α in SNV calling. *F-score=$2 \times \frac{precision \times recall}{precision + recall}$

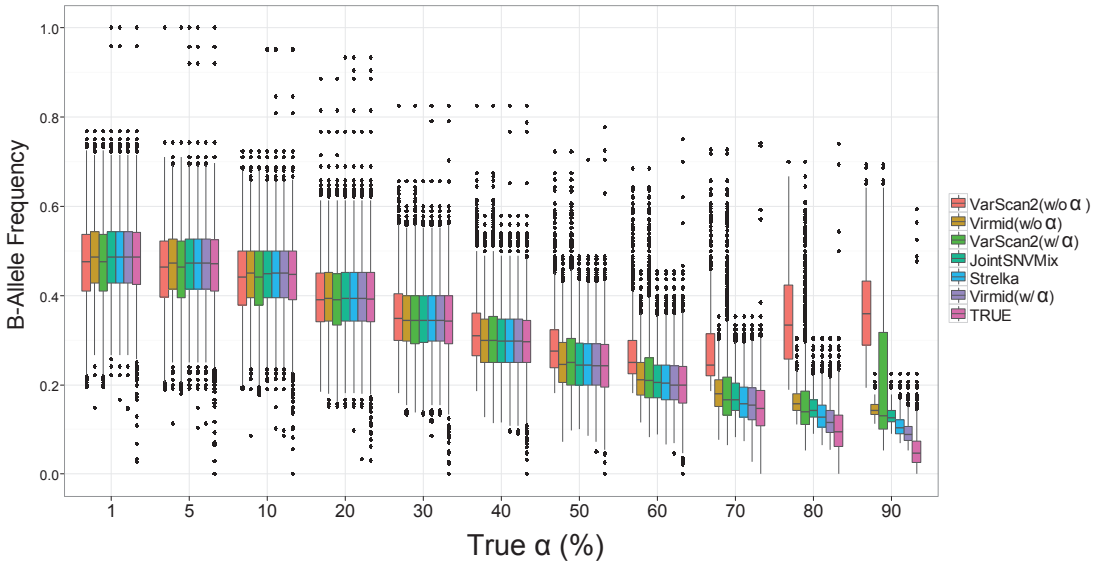


Figure 5.6: BAF distribution of different call-sets. Box plots are drawn for BAF in true (pink boxes) and called somatic mutations. From low to high contamination, the mean BAF decreases from 50% to 5%. Due to the difficulty in finding low BAF mutations, the call-sets show a slight to significant increase in BAF. Virmid with α represents the closest BAF to that of the true set. Due to the undetectable true somatic mutations that contain no `B` alleles, there can be a large gap between true and call set BAF distribution (α=80% and 90%)

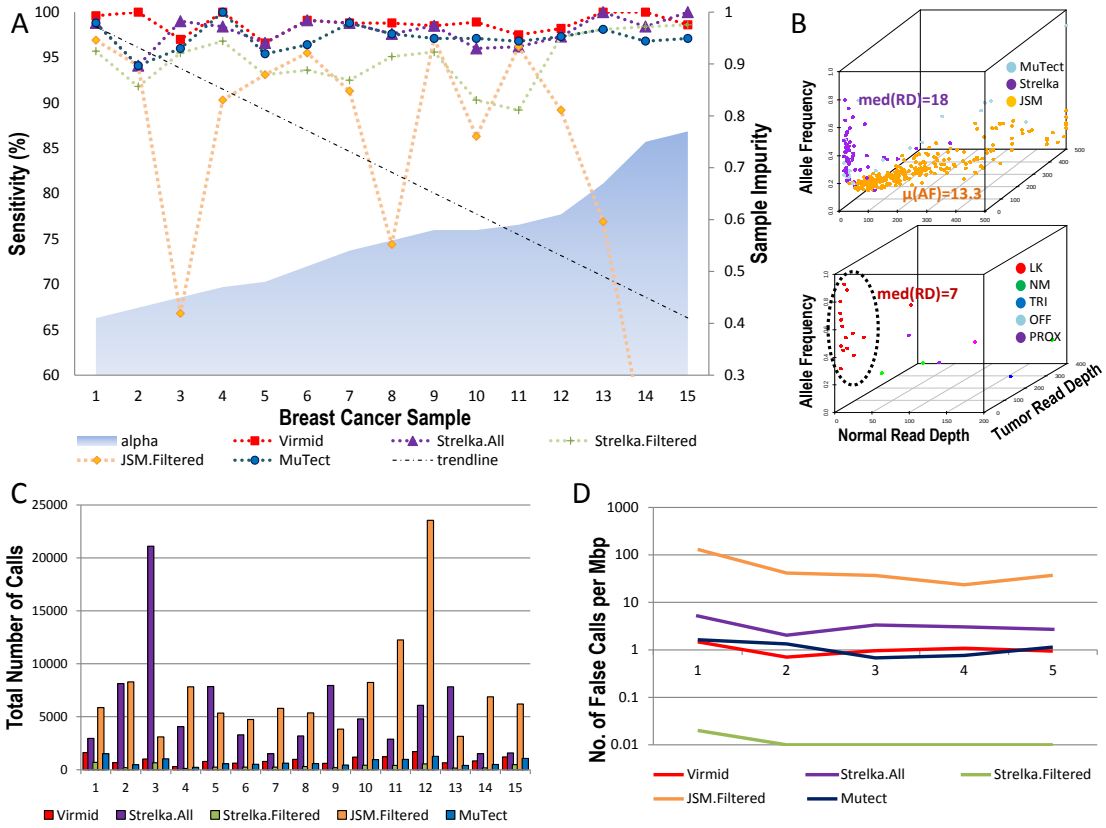


Figure 5.7: **(a)** Sensitivity results of 4 different tools (5 different modes). The samples have been ordered by our estimated α. **(b-upper)** Negative calls from Strelka (purple), MuTect (light blue) and JointSNVMix (orange). Each false negative resulted from low allele frequency or low read depth. **(b-lower)** Many of false negatives from Virmid (15/23) resulted from extremely low coverage in normal sample, which causes insufficient likelihood to call somatic mutation (LK, red dots). Remaining 8 false negatives are explained in filtering error. NM=number of mismatches, TRI=tri-allele, OFF=read offset, and PROX=proximity to indel (see Methods for more details of filtering). **(c)** Total number of calls. **(d)** Specificity result from "virtual tumor" analysis. The number of false positive calls were normalized by the total size of exome region to calculate number of false positives per million base-pair. Note that the y-axis is log-scaled.

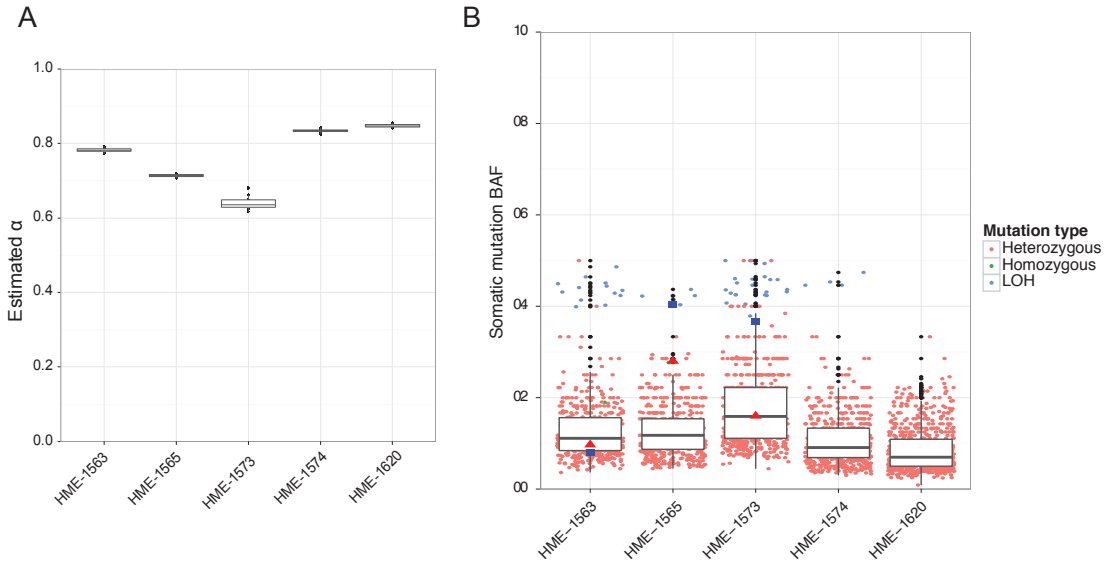


Figure 5.8: Analysis of five hemimegalencephaly (HME) samples. (a) Estimated α values are denoted in boxplot. (b) BAF distributions of HME call sets. Each point represents one predicted somatic mutation in the corresponding sample. BAF is calculated from predicted heterozygous (orange), homozygous (green) somatic mutations and loss of heterozygosity (LOH) sites. The mean BAF is consistent with the estimated α for every sample. The BAF calculated from read counts (red triangles) and mass spectrometry peak intensity (blue squares) of three previously validated mutations (MTOR, AKT3 and PI3KCA) are bounded in the predicted BAF ranges.

Bibliography

[BBL+12] Christopher E Barbieri, Sylvan C Baca, Michael S Lawrence, Francesca Demichelis, Mirjam Blattner, Jean-Philippe Theurillat, Thomas A White, Petar Stojanov, Eliezer Van Allen, Nicolas Stransky, Elizabeth Nickerson, Sung-Suk Chae, Gunther Boysen, Daniel Auclair, Robert C Onofrio, Kyung Park, Naoki Kitabayashi, Theresa Y MacDonald, Karen Sheikh, Terry Vuong, Candace Guiducci, Kristian Cibulskis, Andrey Sivachenko, Scott L Carter, Gordon Saksena, Douglas Voet, Wasay M Hussain, Alex H Ramos, Wendy Winckler, Michelle C Redman, Kristin Ardlie, Ashutosh K Tewari, Juan Miguel Mosquera, Niels Rupp, Peter J Wild, Holger Moch, Colm Morrissey, Peter S Nelson, Philip W Kantoff, Stacey B Gabriel, Todd R Golub, Matthew Meyerson, Eric S Lander, Gad Getz, Mark A Rubin, and Levi A Garraway. Exome sequencing identifies recurrent SPOP, FOXA1 and MED12 mutations in prostate cancer. *Nat Genet*, 44(6):685–689, June 2012.

[BCG07] Marshall Bern, Yuhan Cai, and David Goldberg. Lookup peaks: a hybrid of de novo sequencing and database search for protein identification by tandem mass spectrometry. *Anal. Chem.*, 79(4):1393–1400, 2007.

[BDK+10] Nadine Borchert, Christoph Dieterich, Karsten Krug, Wolfgang Schtz, Stephan Jung, Alfred Nordheim, Ralf J Sommer, and Boris Macek. Proteogenomics of pristionchus pacificus reveals distinct proteome structure of nematode models. *Genome research*, 20(6):837–846, June 2010. PMID: 20237107.

[BGG+08] Katja Baerenfaller, Jonas Grossmann, Monica A. Grobei, Roger Hull, Matthias Hirsch-Hoffmann, Shaul Yalovsky, Philip Zimmermann, Ueli Grossniklaus, Wilhelm Gruissem, and Sacha Baginsky. Genome-scale proteomics reveals arabidopsis thaliana gene models and proteome dynamics. *Science*, 320(5878):938–941, May 2008. PMID: 18436743.

[BGMY04] Marshall Bern, David Goldberg, W Hayes McDonald, and 3rd Yates, John R. Automatic quality assessment of peptide tandem mass spectra. *Bioinformatics (Oxford, England)*, 20 Suppl 1:i49–54, 2004.

[BHH74] E. K. Berndt, B. H. Hall, and R. E. Hall. Estimation and inference in nonlinear structural models. Technical report, National Bureau of Economic Research, Inc, 1974.

[BK11] Marshall Bern and Yong Kil. Comment on Unbiased statistical analysis for Multi-Stage proteomic search strategies. *J. Proteome Res.*, 10(4):2123–2127, 2011.

[BOMP08] Nuno Bandeira, Jesper V. Olsen, Matthias Mann, and Pavel A. Pevzner. Multi-spectra peptide sequencing and its applications to multistage mass spectrometry. *Bioinformatics*, 24(13):i416–i423, January 2008.

[BPG09] Marshall Bern, Brett S Phinney, and David Goldberg. Reanalysis of tyrannosaurus rex mass spectra. *J Proteome Res*, 8(9):4328–32, Sep 2009.

[BSS05] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Inc., 2005.

[BTYW03] Linda A Breci, David L Tabb, 3rd Yates, John R, and Vicki H Wysocki. Cleavage n-terminal to proline: analysis of a database of peptide tandem mass spectra. *Analytical Chemistry*, 75(9):1963–1971, 2003.

[BW09] Sheila J Barton and John C Whittaker. Review of factors that influence the abundance of ions produced in a tandem mass spectrometer and statistical methods for discovering these factors. *Mass Spectrometry Reviews*, 28(1):177–187, 2009.

[Can12] Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, October 2012. PMID: 23000897.

[CB03] Robertson Craig and Ronald C Beavis. A method for reducing the time required to match protein sequences with tandem mass spectra. *Rapid Commun Mass Spectrom*, 17(20):2310–6, Jan 2003.

[CB04] Robertson Craig and Ronald C Beavis. Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–7, Jun 2004.

[CBCC01] J S Choudhary, W P Blackstock, D M Creasy, and J S Cottrell. Interrogating the human genome using uninterpreted mass spectrometry data. *Proteomics*, 1(5):651–667, May 2001. PMID: 11678035.

[CCH+12] Scott L Carter, Kristian Cibulskis, Elena Helman, Aaron McKenna, Hui Shen, Travis Zack, Peter W Laird, Robert C Onofrio, Wendy

Winckler, Barbara A Weir, Rameen Beroukhim, David Pellman, Douglas A Levine, Eric S Lander, Matthew Meyerson, and Gad Getz. Absolute quantification of somatic DNA alterations in human cancer. *Nat Biotech*, advance on, April 2012.

[CFN08] Hyungwon Choi, Damian Fermin, and Alexey I Nesvizhskii. Significance analysis of spectral count data in label-free shotgun proteomics. *Mol Cell Proteomics*, 7(12):2373–85, Dec 2008.

[CKT+01] T Chen, M Y Kao, M Tepel, J Rush, and G M Church. A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, 8(3):325–337, 2001.

[CLC+13] K. Cibulskis, M. S. Lawrence, S. L. Carter, A. Sivachenko, D. Jaffe, C. Sougnez, S. Gabriel, M. Meyerson, E. S. Lander, and G. Getz. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nat. Biotechnol.*, 31(3):213–219, Mar 2013.

[CMF+11] Kristian Cibulskis, Aaron McKenna, Tim Fennell, Eric Banks, Mark DePristo, and Gad Getz. ContEst: estimating cross-contamination of human samples in next-generation sequencing data . *Bioinformatics*, 27(18):2601–2602, September 2011.

[CPS+08] Natalie E. Castellana, Samuel H. Payne, Zhouxin Shen, Mario Stanke, Vineet Bafna, and Steven P. Briggs. Discovery and revision of arabidopsis genes by proteogenomics. *Proceedings of the National Academy of Sciences*, 105(52):21034–21038, December 2008. PMID: 19098097.

[CSY+10] Hao Chi, Rui-Xiang Sun, Bing Yang, Chun-Qing Song, Le-Heng Wang, Chao Liu, Yan Fu, Zuo-Fei Yuan, Hai-Peng Wang, Si-Min He, and Meng-Qiu Dong. pNovo: de novo peptide sequencing and identification using HCD spectra. *J. Proteome Res.*, 9(5):2713–2724, 2010.

[DAC+99] V Dancik, T A Addona, K R Clauser, J E Vath, and P A Pevzner. De novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 6(3-4):327–342, 1999.

[DB09] Ritendra Datta and Marshall Bern. Spectrum fusion: using multiple mass spectra for de novo peptide sequencing. *Journal of Computational Biology*, 16(8):1169–1182, 2009.

[DBP+11] M. A. DePristo, E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. A. Philippakis, G. del Angel, M. A. Rivas, M. Hanna, A. McKenna, T. J. Fennell, A. M. Kernytsky, A. Y. Sivachenko, K. Cibulskis, S. B. Gabriel, D. Altshuler, and M. J. Daly. A framework

for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.*, 43(5):491–498, May 2011.

[EBM10] Logan J. Everett, Charlene Bierl, and Stephen R. Master. Unbiased statistical analysis for Multi-Stage proteomic search strategies. *J. Proteome Res.*, 9(2):700–707, 2010.

[ECL+12] G. D. Evrony, X. Cai, E. Lee, L. B. Hills, P. C. Elhosary, H. S. Lehmann, J. J. Parker, K. D. Atabay, E. C. Gilmore, A. Poduri, P. J. Park, and C. A. Walsh. Single-neuron sequencing analysis of L1 retrotransposition and somatic mutation in the human brain. *Cell*, 151(3):483–496, Oct 2012.

[EG07] Joshua E Elias and Steven P Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods*, 4(3):207–14, 2007.

[EG10] Joshua E Elias and Steven P Gygi. Target-decoy search strategy for mass spectrometry-based proteomics. *Methods Mol Biol*, 604:55–71, Jan 2010.

[EMY94] Jimmy K. Eng, Ashley L. McCormack, and John R. 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*, 5(11):976–989, 1994.

[Epp98] D. Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673, 1998.

[FAH+11] Christian K. Frese, A. F. Maarten Altelaar, Marco L. Hennrich, Dirk Nolting, Martin Zeller, Jens Griep-Raming, Albert J. R. Heck, and Shabaz Mohammed. Improved peptide identification by targeted fragmentation using CID, HCD and ETD on an LTQ-Orbitrap velos. *J. Proteome Res.*, 10(5):2377–2388, 2011.

[FBS+08] Ari M Frank, Nuno Bandeira, Zhouxin Shen, Stephen Tanner, Steven P Briggs, Richard D Smith, and Pavel A Pevzner. Clustering millions of tandem mass spectra. *Journal of proteome research*, 7(1):113–122, January 2008. PMID: 18067247.

[Fis] R. A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, (1):87–94.

[FP05] Ari Frank and Pavel Pevzner. PepNovo: de novo peptide sequencing via probabilistic network modeling. *Anal. Chem.*, 77(4):964–973, 2005.

[Fra09] Ari Frank. A ranking-based scoring function for peptide-spectrum matches. *Journal of Proteome Research*, 8(5):2241–2252, 2009.

[GMK+04] Lewis Y Geer, Sanford P Markey, Jeffrey A Kowalak, Lukas Wagner, Ming Xu, Dawn M Maynard, Xiaoyu Yang, Wenyao Shi, and Stephen H Bryant. Open mass spectrometry search algorithm. *Journal of proteome research*, 3(5):958–964, October 2004. PMID: 15473683.

[GNK] Viktor Granholm, William Stafford Noble, and Lukas Käll. On using samples of known protein content to assess the statistical calibration of scores assigned to Peptide-Spectrum matches in shotgun proteomics. *J. Proteome Res.*, (5):2671–2678.

[GP09] Nitin Gupta and Pavel A. Pevzner. False discovery rates of protein identifications: A strike against the two-peptide rule. *Journal of Proteome Research*, 8(9):4173–4181, September 2009. PMID: 19627159 PMCID: PMC3398614.

[GSM+10] Rodrigo Goya, Mark G.F. Sun, Ryan D. Morin, Gillian Leung, Gavin Ha, Kimberley C. Wiegand, Janine Senz, Anamaria Crisan, Marco A. Marra, Martin Hirst, David Huntsman, Kevin P. Murphy, Sam Aparicio, and Sohrab P. Shah. Snvmix: predicting single nucleotide variants from next-generation sequencing of tumors. *Bioinformatics*, 26(6):730–736, 2010.

[GTJ+07] Nitin Gupta, Stephen Tanner, Navdeep Jaitly, Joshua N Adkins, Mary Lipton, Robert Edwards, Margaret Romine, Andrei Osterman, Vineet Bafna, Richard D Smith, and Pavel A Pevzner. Whole proteome analysis of post-translational modifications: applications of mass-spectrometry for proteogenomic annotation. *Genome research*, 17(9):1362–1377, September 2007. PMID: 17690205.

[Gus97] Dan Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* Cambridge University Press, 1 edition, May 1997.

[HJP+01] L Huang, R J Jacob, S C Pegg, M A Baldwin, C C Wang, A L Burlingame, and P C Babbitt. Functional assignment of the 20 s proteasome from trypanosoma brucei using mass spectrometry and new bioinformatics approaches. *The Journal of biological chemistry*, 276(30):28327–28339, July 2001. PMID: 11309374.

[HM10] Lin He and Bin Ma. ADEPTS: advanced peptide de novo sequencing with a pair of tandem mass spectra. *Journal of Bioinformatics and Computational Biology*, 8(6):981–994, 2010.

[HTT+05] Yingying Huang, Joseph M Triscari, George C Tseng, Ljiljana Pasa-Tolic, Mary S Lipton, Richard D Smith, and Vicki H Wysocki. Statistical characterization of the charge state and residue dependence of

low-energy CID peptide dissociation patterns. *Analytical Chemistry*, 77(18):5800–5813, 2005.

[Hun76] David Hunter. An upper bound for the probability of a union. *Journal of Applied Probability*, 13(3):597–603, 1976.

[JKBP11] Kyowon Jeong, Sangtae Kim, Nuno Bandeira, and Pavel A. Pevzner. Gapped spectral dictionaries and their applications for database searches of tandem mass spectra. *Molecular & Cellular Proteomics*, 10(6):M110.002220, 2011.

[JMB+87] Richard S. Johnson, Stephen A. Martin, Klaus Biemann, John T. Stults, and J. Throck Watson. Novel fragmentation process of peptides by collision-induced decomposition in a tandem mass spectrometer: differentiation of leucine and isoleucine. *Anal. Chem.*, 59(21):2621–2625, 1987.

[JT02] Richard S. Johnson and J. Alex Taylor. Searching sequence databases via de novo peptide sequencing by tandem mass spectrometry. *Molecular Biotechnology*, 22(3):301–315, November 2002.

[Kay93] Steven M. Kay. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[KBP09] Sangtae Kim, Nuno Bandeira, and Pavel A Pevzner. Spectral profiles, a novel representation of tandem mass spectra and their applications for de novo peptide sequencing and identification. *Molecular & Cellular Proteomics*, 8(6):1391–1400, 2009.

[KCW+07] Lukas Käll, Jesse D. Canterbury, Jason Weston, William Stafford Noble, and Michael J. MacCoss. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods*, 4(11):923–925, 2007.

[KCW+09] Daniel C. Koboldt, Ken Chen, Todd Wylie, David E. Larson, Michael D. McLellan, Elaine R. Mardis, George M. Weinstock, Richard K. Wilson, and Li Ding. Varscan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics*, 25(17):2283–2285, 2009.

[KDW+04] Paul J Kersey, Jorge Duarte, Allyson Williams, Youla Karavidopoulou, Ewan Birney, and Rolf Apweiler. The international protein index: an integrated database for proteomics experiments. *Proteomics*, 4(7):1985–1988, 2004.

[KEH+08] John Klimek, James S. Eddes, Laura Hohmann, Jennifer Jackson, Amelia Peterson, Simon Letarte, Philip R. Gafken, Jonathan E

Katz, Parag Mallick, Hookeun Lee, Alexander Schmidt, Reto Ossola, Jimmy K. Eng, Ruedi Aebersold, and Daniel B Martin. The standard protein mix database: A diverse data set to assist in the production of improved peptide and protein identification software tools. *Journal of Proteome Research*, 7(1):96–103, 2008.

[KGBP09] Sangtae Kim, Nitin Gupta, Nuno Bandeira, and Pavel A. Pevzner. Spectral dictionaries. *Molecular & Cellular Proteomics*, 8(1):53 –69, 2009.

[KGP08] Sangtae Kim, Nitin Gupta, and Pavel A. Pevzner. Spectral probabilities and generating functions of tandem mass spectra: A strike against decoy databases. *Journal of Proteome Research*, 7(8):3354–3363, August 2008.

[KMAM01] B Kster, P Mortensen, J S Andersen, and M Mann. Mass spectrometry allows direct identification of proteins in large genomes. *Proteomics*, 1(5):641–650, May 2001. PMID: 11678034.

[KMB+10] Sangtae Kim, Nikolai Mischerikow, Nuno Bandeira, J. Daniel Navarro, Louis Wich, Shabaz Mohammed, Albert J. R. Heck, and Pavel A. Pevzner. The generating function of CID, ETD, and CID/ETD pairs of tandem mass spectra: Applications to database search. *Molecular & Cellular Proteomics*, 9(12):2840 –2852, 2010.

[KNKA02] A Keller, A Nesvizhskii, E Kolker, and R Aebersold. Empirical statistical model to estimate the accuracy of peptide identifications made by ms/ms and database search. *Anal. Chem.*, 74:5383–92, Jan 2002.

[KSMN08] Lukas Käll, John D Storey, Michael J Maccoss, and William Stafford Noble. Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *J Proteome Res*, 7(1):29–34, Jan 2008.

[KSV+12] Jacob O Kitzman, Matthew W Snyder, Mario Ventura, Alexandra P Lewis, Ruolan Qiu, LaVone E Simmons, Hilary S Gammill, Craig E Rubens, Donna A Santillan, Jeffrey C Murray, Holly K Tabor, Michael J Bamshad, Evan E Eichler, and Jay Shendure. Noninvasive Whole-Genome Sequencing of a Human Fetus . *Science Translational Medicine*, 4(137):137ra76–137ra76, June 2012.

[KZL+12] Daniel C. Koboldt, Qunyuan Zhang, David E. Larson, Dong Shen, Michael D. McLellan, Ling Lin, Christopher A. Miller, Elaine R. Mardis, Li Ding, and Richard K. Wilson. Varscan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Research*, 22(3):568–576, 2012.

[LD10] Heng Li and Richard Durbin. Fast and accurate long-read alignment with burrows-wheeler transform. *Bioinformatics*, 26(5):589–595, mar. 2010.

[LDA10] H Lam, E W Deutsch, and R Aebersold. Artificial decoy spectral libraries for false discovery rate estimation in spectral library searching in proteomics. *J Proteome Res*, 9(1):605–610, Jan 2010.

[LHC+12] David E. Larson, Christopher C. Harris, Ken Chen, Daniel C. Koboldt, Travis E. Abbott, David J. Dooling, Timothy J. Ley, Elaine R. Mardis, Richard K. Wilson, and Li Ding. Somaticsniper: identification of somatic point mutations in whole genome sequencing data. *Bioinformatics*, 28(3):311–317, 2012.

[LHS+12] J. H. Lee, M. Huynh, J. L. Silhavy, S. Kim, T. Dixon-Salazar, A. Heiberg, E. Scott, V. Bafna, K. J. Hill, A. Collazo, V. Funari, C. Russ, S. B. Gabriel, G. W. Mathern, and J. G. Gleeson. De novo somatic mutations in components of the PI3K-AKT3-mTOR pathway cause hemimegalencephaly. *Nat Genet*, Jun 2012.

[LHW+09] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[Li11] Heng Li. Tabix: fast retrieval of sequence features from generic tab-delimited files. *Bioinformatics*, 27(5):718–719, 2011.

[LS12] H. Lee and M. C. Schatz. Genomic Dark Matter: The reliability of short read mapping illustrated by the Genome Mappability Score. *Bioinformatics*, Jun 2012.

[LSN+07] Samuel Levy, Granger Sutton, Pauline C Ng, Lars Feuk, Aaron L Halpern, Brian P Walenz, Nelson Axelrod, Jiaqi Huang, Ewen F Kirkness, Gennady Denisov, Yuan Lin, Jeffrey R MacDonald, Andy Wing Chun Pang, Mary Shago, Timothy B Stockwell, Alexia Tsiamouri, Vineet Bafna, Vikas Bansal, Saul A Kravitz, Dana A Busam, Karen Y Beeson, Tina C McIntosh, Karin A Remington, Josep F Abril, John Gill, Jon Borman, Yu-Hui Rogers, Marvin E Frazier, Stephen W Scherer, Robert L Strausberg, and J. Craig Venter. The diploid genome sequence of an individual human. *PLoS Biol*, 5(10):e254, 09 2007.

[LSXM10] Xiaowen Liu, Baozhen Shan, Lei Xin, and Bin Ma. Better score function for peptide identification with ETD MS/MS spectra. *BMC Bioinformatics*, 11(Suppl 1):S4, 2010.

[MHB+10] Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, and Mark A. DePristo. The genome analysis toolkit: A mapreduce framework for analyzing next-generation dna sequencing data. *Genome Research*, 20(9):1297–1303, 2010.

[MJ11] Bin Ma and Richard Johnson. De novo sequencing and homology searching. *Molecular & Cellular Proteomics*, page O111.014902, 2011.

[MK08] Matthias Mann and Neil L Kelleher. Precision proteomics: the case for high resolution and high mass accuracy. *Proc Natl Acad Sci USA*, 105(47):18132–8, Nov 2008.

[MLT12] Kerensa E McElroy, Fabio Luciani, and Torsten Thomas. Gemsim: general, error-model based simulator of next-generation sequencing data. *BMC genomics*, 13(1):74, feb. 2012.

[MUD+08] Hiroshi Makino, Hiroyuki Uetake, Kathleen Danenberg, Peter Danenberg, and Kenichi Sugihara. Efficacy of laser capture microdissection plus rt-pcr technique in analyzing gene expression levels in human gastric cancer and colon cancer. *BMC Cancer*, 8(1):210, 2008.

[MW94] M Mann and M Wilm. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Analytical chemistry*, 66(24):4390–4399, December 1994. PMID: 7847635.

[MZH+03] Bin Ma, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby, and Gilles Lajoie. PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry: RCM*, 17(20):2337–2342, 2003.

[NAP11] Julio Ng, Amihood Amir, and Pavel A. Pevzner. Blocked pattern matching problem and its applications in proteomics. RECOMB 2011, Vancouver, Canada, pages 298–319, 2011.

[Nes10] Alexey I Nesvizhskii. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *J Proteomics*, 73(11):2092–123, 2010.

[NP06] Seungjin Na and Eunok Paek. Quality assessment of tandem mass spectra based on cumulative intensity normalization. *J Proteome Res*, 5(12):3241–8, Dec 2006.

[NZVLW+12] Serena Nik-Zainal, Peter Van Loo, David C Wedge, Ludmil B Alexandrov, Christopher D Greenman, King Wai Lau, Keiran Raine, David Jones, John Marshall, Manasa Ramakrishna, Adam Shlien, Susanna L

Cooke, Jonathan Hinton, Andrew Menzies, Lucy A Stebbings, Catherine Leroy, Mingming Jia, Richard Rance, Laura J Mudie, Stephen J Gamble, Philip J Stephens, Stuart McLaren, Patrick S Tarpey, Elli Papaemmanuil, Helen R Davies, Ignacio Varela, David J McBride, Graham R Bignell, Kenric Leung, Adam P Butler, Jon W Teague, Sancha Martin, Goran Jnsson, Odette Mariani, Sandrine Boyault, Penelope Miron, Aquila Fatima, Anita Langerd, Samuel A J R Aparicio, Andrew Tutt, Anieta M Sieuwerts, ke Borg, Gilles Thomas, Anne Vincent Salomon, Andrea L Richardson, Anne-Lise Brresen-Dale, P Andrew Futreal, Michael R Stratton, Peter J Campbell, and Breast Cancer Working Group of the International Cancer Genome Consortium. The life history of 21 breast cancers. *Cell*, 149(5):994–1007, May 2012. PMID: 22608083.

[OML+07] Jesper V Olsen, Boris Macek, Oliver Lange, Alexander Makarov, Stevan Horning, and Matthias Mann. Higher-energy c-trap dissociation for peptide modification analysis. *Nature Methods*, 4(9):709–712, 2007.

[OWW+02] Guy Oshiro, Lisa M. Wodicka, Michael P. Washburn, John R. Yates, David J. Lockhart, and Elizabeth A. Winzeler. Parallel identification of new genes in saccharomyces cerevisiae. *Genome Research*, 12(8):1210–1220, August 2002. PMID: 12176929 PMCID: PMC186640.

[PBH+] Amanda G Paulovich, Dean Billheimer, Amy-Joan L Ham, Lorenzo Vega-Montoto, Paul A Rudnick, David L Tabb, Pei Wang, Ronald K Blackman, David M Bunk, Helene L Cardasis, Karl R Clauser, Christopher R Kinsinger, Birgit Schilling, Tony J Tegeler, Asokan Mulayath Variyath, Mu Wang, Jeffrey R Whiteaker, Lisa J Zimmerman, David Fenyo, Steven A Carr, Susan J Fisher, Bradford W Gibson, Mehdi Mesri, Thomas A Neubert, Fred E Regnier, Henry Rodriguez, Cliff Spiegelman, Stephen E Stein, Paul Tempst, and Daniel C Liebler. Interlaboratory study characterizing a yeast performance standard for benchmarking LC-MS platform performance. *Molecular & Cellular Proteomics*, (2):242–254.

[PPCC99] D N Perkins, D J Pappin, D M Creasy, and J S Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18):3551–3567, 1999.

[RMD+12] Andrew Roth, Ryan Morin, Jiarui Ding, Anamaria Crisan, Gavin Ha, Ryan Giuliany, Ali Bashashati, Martin Hirst, Gulisa Turashvili, Arusha Oloumi, Marco A Marra, Samuel Aparicio, and Sohrab P Shah. JointSNVMix : A Probabilistic Model For Accurate Detection Of Somatic Mutations In Normal/Tumour Paired Next Generation Sequencing Data . *Bioinformatics*, January 2012.

[SASK12] Yuki Sugaya, Yasuaki Akazawa, Akira Saito, and Shigeo Kamitsuji. NDesign: software for study design for the detection of rare variants from next-generation sequencing data. *J Hum Genet*, July 2012.

[SBP+05] Adam Siepel, Gill Bejerano, Jakob S. Pedersen, Angie S. Hinrichs, Minmei Hou, Kate Rosenbloom, Hiram Clawson, John Spieth, LaDeana W. Hillier, Stephen Richards, George M. Weinstock, Richard K. Wilson, Richard A. Gibbs, W. James Kent, Webb Miller, and David Haussler. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Research*, 15(8):1034–1050, 2005.

[SDW+05] Brian C Searle, Surendra Dasari, Phillip A Wilmarth, Mark Turner, Ashok P Reddy, Larry L David, and Srinivasa R Nagalla. Identification of protein modifications using MS/MS de novo sequencing and the OpenSea alignment algorithm. *Journal of proteome research*, 4(2):546–554, April 2005. PMID: 15822933.

[SKG+06] Mario Stanke, Oliver Keller, Irfan Gunduz, Alec Hayes, Stephan Waack, and Burkhard Morgenstern. AUGUSTUS: ab initio prediction of alternative transcripts. *Nucleic Acids Research*, 34(suppl 2):W435–W439, July 2006. PMID: 16845043.

[SKVQ11] Thomas M Snyder, Kiran K Khush, Hannah A Valantine, and Stephen R Quake. Universal noninvasive detection of solid organ transplant rejection . *Proceedings of the National Academy of Sciences*, 108(15):6229–6234, April 2011.

[SMC08] Danielle L Swaney, Graeme C McAlister, and Joshua J Coon. Decision tree-driven tandem mass spectrometry for shotgun proteomics. *Nature Methods*, 5(11):959–964, 2008.

[SMG12a] G. Saksena, C. Mermel, and G. Getz. Developing algorithms to discover novel cancer genes: A look at the challenges and approaches. *Signal Processing Magazine, IEEE*, 29(1):89 –97, jan. 2012.

[SMG+12b] S. J. Sanders, M. T. Murtha, A. R. Gupta, J. D. Murdoch, M. J. Raubeson, A. J. Willsey, A. G. Ercan-Sencicek, N. M. DiLullo, N. N. Parikshak, J. L. Stein, M. F. Walker, G. T. Ober, N. A. Teran, Y. Song, P. El-Fishawy, R. C. Murtha, M. Choi, J. D. Overton, R. D. Bjornson, N. J. Carriero, K. A. Meyer, K. Bilguvar, S. M. Mane, N. Sestan, R. P. Lifton, M. Gunel, K. Roeder, D. H. Geschwind, B. Devlin, and M. W. State. De novo mutations revealed by whole-exome sequencing are strongly associated with autism. *Nature*, 485(7397):237–241, May 2012.

[SNKZ05] Mikhail M. Savitski, Michael L. Nielsen, Frank Kjeldsen, and Roman A. Zubarev. Proteomics-Grade de novo sequencing approach. *J. Proteome Res.*, 4(6):2348–2354, 2005.

[Sto] John Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):479–498.

[SWC10] Danielle L. Swaney, Craig D. Wenger, and Joshua J. Coon. Value of using multiple proteases for Large-Scale mass Spectrometry-Based proteomics. *J. Proteome Res.*, 9(3):1323–1329, 2010.

[SWK+01] S. T. Sherry, M. H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29(1):308–311, jan. 2001.

[SWS+12] Christopher T. Saunders, Wendy S. W. Wong, Sajani Swamy, Jennifer Becq, Lisa J. Murray, and R. Keira Cheetham. Strelka: accurate somatic small-variant calling from sequenced tumornormal sample pairs. *Bioinformatics*, 28(14):1811–1817, 2012.

[SZZ+12] Xiaoping Su, Li Zhang, Jianping Zhang, Funda Meric-Bernstam, and John N. Weinstein. Purityest: estimating purity of human tumor samples using next-generation sequencing data. *Bioinformatics*, 2012.

[THWY04] David L. Tabb, Yingying Huang, Vicki H. Wysocki, and John R. Yates. Influence of basic residue content on fragment ion peak intensities in Low-Energy Collision-Induced dissociation spectra of peptides. *Anal. Chem.*, 76(5):1243–1248, 2004.

[TSF+05] Stephen Tanner, Hongjun Shu, Ari Frank, Ling-Chi Wang, Ebrahim Zandi, Marc Mumby, Pavel A. Pevzner, and Vineet Bafna. InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. *Anal. Chem.*, 77(14):4626–4639, 2005.

[TSN+07] Stephen Tanner, Zhouxin Shen, Julio Ng, Liliana Florea, Roderic Guig, Steven P Briggs, and Vineet Bafna. Improving gene annotation using peptide mass spectrometry. *Genome research*, 17(2):231–239, February 2007. PMID: 17189379.

[WTSB00] Vicki H Wysocki, George Tsaprailis, Lori L Smith, and Linda A Breci. Mobile and localized protons: a framework for understanding peptide dissociation. *Journal of Mass Spectrometry*, 35(12):1399–1406, 2000.

[XRD+11] B. Xu, J. L. Roos, P. Dexheimer, B. Boone, B. Plummer, S. Levy, J. A. Gogos, and M. Karayiorgou. Exome sequencing supports a de novo mutational paradigm for schizophrenia. *Nat. Genet.*, 43(9):864–868, Sep 2011.

[YEM95] 3rd Yates, J R, J K Eng, and A L McCormack. Mining genomes: correlating tandem mass spectra of modified and unmodified peptides to sequences in nucleotide databases. *Analytical chemistry*, 67(18):3202–3210, September 1995. PMID: 8686885.

[YMJ+10] Christopher Yau, Dmitri Mouradov, Robert Jorissen, Stefano Colella, Ghazala Mirza, Graham Steers, Adrian Harris, Jiannis Ragoussis, Oliver Sieber, and Christopher Holmes. A statistical approach for detecting genomic aberrations in heterogeneous tumor samples from single nucleotide polymorphism genotyping data. *Genome Biology*, 11(9):R92, 2010.

[ZZS08] Roman A. Zubarev, Alexander R. Zubarev, and Mikhail M. Savitski. Electron Capture/Transfer versus collisionally Activated/Induced dissociations: Solo or duet? *Journal of the American Society for Mass Spectrometry*, 19:753–761, 2008.