NATIONAL RESEARCH UNIVERSITY HIGHER SCHOOL OF ECONOMICS

as a manuscript

Attila Kertesz-Farkas

Computational methods for tandem mass spectrometry data annotation

Dissertation Summary

for the purpose of obtaining academic degree
Doctor of Science in Computer Science

Moscow — 2022

1 Introduction

This thesis dissertation presents computational and statistical methods which are used to annotate spectra obtained with a tandem mass spectrometer. A spectrum consists of several peaks, each peak has an associated (1) real-valued location in mass-to-charge units, denoted as m/z sometimes simply (but always incorrectly) referred to as mass, and (2) an intensity value indicating the height of the peak. An example of a spectrum is illustrated in Figure 1, the spikes represent peaks. The spectrum obtained from spectrometer instruments is referred to as an experimental or an observed spectrum to distinguish from the so-called theoretical spectrum that will be introduced later. A typical experimental spectrum contains around from few-tens to several hundreds of peaks, and a typical experiments provides hundred of thousands or millions of spectra to be annotated. An experimental spectrum can be considered as a fingerprint of a molecule, which are yet to be identified [1, 2].

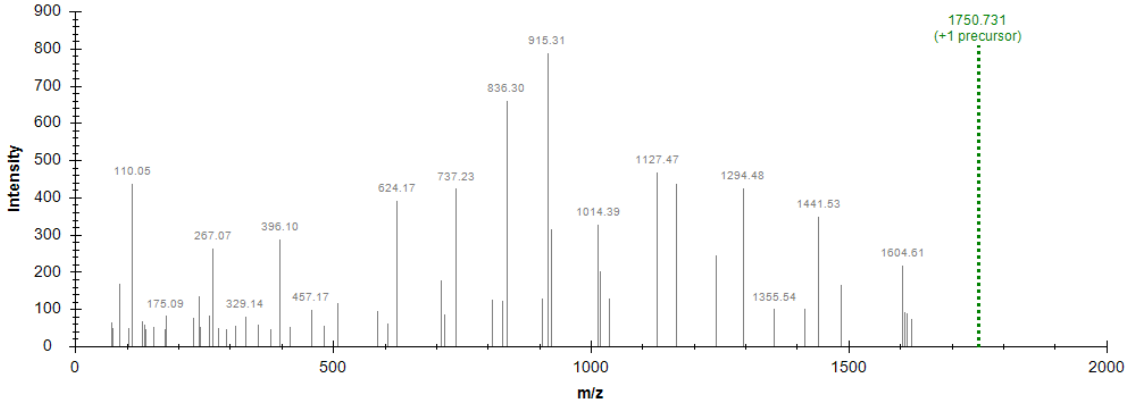

Figure 1: An illustration of an experimental spectrum. A mass spectrometer generates an experimental spectrum from a chemical molecule. The main task is to identify the molecule from the spectrum.

The main question of this thesis is stated as follows. What is the molecule which is responsible for generating this experimental spectrum observed by the spectrometer? Perhaps more interesting questions are why and how likely is this annotation correct? These are the central questions of this thesis.

More specifically, this thesis focuses on methods to identify peptide[1] molecules in biological sample, such as blood or tissue samples in proteomics studies. The de facto way here to annotate experimental spectrum is based on database-searching. In this approach an experimental spectrum s is iteratively compared and scored against a large database of reference peptides h_js. The experimental spectrum s is annotated by the best-scoring reference peptide $\hat{h}$. This can be formalized as follows:

$$s \leftarrow \hat{h} = \underset{h_j \in CP(s)}{\arg\max}\ \phi(s, h_j), \tag{1}$$

where $s \leftarrow \hat{h}$ means that the observed spectrum s is annotated with the peptide sequence h from the reference data set. This consists of three key elements: (1) the peptide database DB, (2) a

[1]Peptide is a short chain of amino acids. Proteins are large peptides.

selection of biologically/chemically plausible peptides, called candidate peptides ($CP(s) \subseteq DB$) with respect to an experimental spectrum s, and (3) the score function $\phi : S \times DB \rightarrow R$, where S denotes the set of spectra obtained from an experiment and R denotes the real numbers. The elements and the size of the CP highly varies for different experimental spectra. The scoring typically provides a similarity-like score (i.e. higher score indicates a better match) based on matching the peaks of the experimental to reference peaks generated from the peptide sequences in silico [3].

This seems to be an easy task and a problem solved, what are the challenges here? Well, the main problem is that it cannot be guaranteed that the best scoring peptide does in fact give a correct annotation. Roughly, the 60-80% of the experimental spectra cannot be annotated correctly with high confidence. The main challenges which hamper spectrum annotation are the following:

1. *Detector inaccuracy.* The location of the observed peaks (m/z) in the experimental spectra is not entirely accurate due to the inaccuracy of the detector in the spectrometer. This leads to an uncertainty for the score functions when matching inexact observed peaks to exact peaks of the peptides in the reference data set (DB). "Old" spectrometers with detectors of low resolution can distinguish between the mass of the proton, while "modern" spectrometers with detectors of high resolution can distinguish between the one-fiftieth (1/50) of the mass of the proton. Score functions can take advantage of the higher degree of granularity provided by detectors of high-resolution.

2. *Discriminative power of score functions.* It means the ability of the score function to distinguish between the correct and incorrect peptide-spectrum-matches (PSMs). Score functions are hindered by (a) the presence of many unexplainable peaks, which stem from the unusual fragmentation of the peptide or contaminating molecules, or (b) by the lack of expected fragmentation ions, which fail to be observed in the mass spectrometer.

3. *Calibration of score functions.* Uncalibrated, raw PSM scores may indicate different match quality for different spectra. For instance, a raw score of 2.5 may imply a correct annotation for a spectrum obtained from a, say, small peptide molecule but it may imply an incorrect annotation for a spectrum obtained from, say, a large peptide molecule [4]. Spectrum-specific score calibration methods aim to provide a sort of score normalization so that spectrum assignments become comparable with each other; therefore, a single threshold can be selected to accept or reject spectrum annotations for the whole experiment. The calibration allows one to obtain many more spectrum annotations at any desired false discovery rate (FDR). Score calibration methods involve a null distribution and calibrate a raw score to either the mean or the tail of the null distribution [5].

4. *The content of the CP set.* A spectrum cannot be annotated correctly, if the correct peptide sequence is missing from the reference data set. Protein and peptide molecules often undergo certain chemical or post-translation modification (PTM) which changes the mass and/or the composition of the molecules. Other times, the biological sample preparation ends up with

unwanted modification to the sample. To overcome this issue, the possible modification needs to be considered during the CP generation step. However, one might ask why do not we just generate all the possible amino acid sequences with all the possible modifications to make sure that a spectrum will be annotated? The next point answers this question.

5. *The size of the CP set.* A spectrum is annotated by the top scoring element of the CP set, and the accompanying best score undergoes a sort of multiple testing correction. Thus, a high score may not end up being statistically significant. Employing too large CP sets involves too strong correction factors which in turn results in fewer number of annotations with high confidence than what we could obtain using a smaller CP set. Thus, it is essential to ensure that the CP set contains the possibly correct peptide sequences but the CP set is not too large that reduces the statistical confidence values [6].

The research results presented here are methods to increase the number of spectrum annotations with high statistical confidence.

1.1 The relevance of research

Mass spectrometry is the de facto method to identify molecules in a mixture of samples in several disciplines including molecular biology, forensic, pharmaceutical industry, medicine, etcetera. For instance, in environmental contamination analysis the mass spectrometry can be used to test food and beverages for contamination or adulteration. Soil analysis can be carried out with mass spectrometers to estimate the amount of the pesticides or hormone used in cultivation. In forensics analysis, mass spectrometry can be used to confirm drug abuse or identify explosive residues or fire accelerants to determine incendiarism. In pharmaceutical analysis, determining structures of drugs and metabolites, as well as screening for metabolites in biological systems are the main applications of mass-spectrometry analysis. In clinical researches and clinical drug development the mass spectrometer is used in disease screening, drug therapy monitoring to monitor protein composition of cells in study, and identification of infectious agents for targeted therapies.

Accurate data identification and spectrum annotations are essential for experimenters and practitioners working on the fields mentioned above.

1.2 Importance of work

Single experiment may require weeks or month of sample preparation and hundred of hours of human labor force. It also may require expensive materials, compounds, and instruments. Hence, an experiment can be time consuming and it may cost thousands of dollars. Therefore, accurate data annotation is essential for experimenters and practitioners working with mass spectrometers in order to conclude correct conclusions about their experiments and to make proper decisions for future experiments or clinical therapies, for instance, in selecting the right drug therapy. Therefore, it is important to develop reliable and accurate methods to annotate and identify spectra with high confidence for data obtained with various types of spectrometers using various experimental

protocols and sample preparation methods.

1.3 Novelty and summary of the Author's main results

The main results of this thesis are computational methods and statistical protocols in order to improve the number for spectrum annotation annotated with high confidence. The results can be categorized in few main areas as follows. The publications 1, 8, 12 from Table 1 present new spectrum scoring methods with improved discriminative power. The publications 2, 3, 4, 9 present statistical methods for score calibration or statistical protocols with increased statistical power in spectrum annotation. The publications 5 and 6 present methods which find post-translational modifications in the spectrum data. The publications 7 and 11 present spectrum filtering methods. Finally, the publication 10 presents an open source toolkit of analysis tools for interpreting mass spectrometry data, and the publications 13 and 14 are two review papers on database-searching approach and spectrum filtering methods, respectively.

1.4 Publications

This dissertation is based on a collection of 14 articles listed in Table 1, all are in Scopus or Core A*, A, B venues. The doctoral school of computer science (DSCS) of the HSE University in Moscow, Russia, requires at least 10 articles. Among these 14 articles, 11 are published in Scopus Q1-Q2 or Core A venues (8 are required by DSCS). I am the main co-author of 7 out of these 11 articles (4 are required by DSCS). I coauthored 9 articles with main contributions published in first or second tier venues. During the dissertation defense, I present 7 articles (1-7) (7 are required by DSCS). Therefore, this dissertation meets the publication criteria required by DSCS. None of these articles have been used for my PhD degree. I obtained my PhD in 2010; however, all of these articles have been published after 2010. Therefore, the main results of these articles are not used for obtaining academic degree twice. My other publications not strictly related to computational mass spectrometry are listed in Table 2.

Table 1: Dissertation publications

	Authors, Title, Venue
First-tier publications in Scopus Q1 and Core A venues with main contribution	
1	Polina Kudriavtseva, Matvey Kashkinov, Attila Kertész-Farkas* *Deep convolutional neural networks help scoring tandem mass spectrometry data in database-searching approaches* **Journal of Proteome Research**, 2021, https://doi.org/10.1021/acs.jproteome.1c00315
2	Pavel Sulimov, Attila Kertész-Farkas* *Tailor: non-parametric and rapid score calibration method for database search-based peptide identification in shotgun proteomics* **Journal of Proteome Research**, 2020, 18(5), 2354–2358
3	Yulia Danilova, Anastasia Voronkova, Pavel Sulimov, Attila Kertész-Farkas* *Bias in false discovery rate estimation in mass-spectrometry-based peptide identification* **Journal of Proteome Research**, 2019, 18(5), 2354–2358
4	Attila Kértesz-Farkas, Uri Keich, William Stafford Noble *Tandem Mass Spectrum Identification via Cascaded Search* **Journal of Proteome Research**, 2015, 14(8), 3027–3038
5	Attila Kertész-Farkas, Beáta Reiz, Roberto Vera, Michael P. Myers, Sándor Pongor *PTMTreeSearch: a Novel Two-Stage Tree Search Algorithm with Pruning Rules for the Identification of Post-Translational Modification of Proteins in MS/MS Spectra* **Bioinformatics**, 2014, 30(2), 234–241
6	Attila Kertész-Farkas, Beáta Reiz, Michael Myers, Sándor Pongor *PTMSearch: A Greedy Tree Traversal Algorithm for Finding Protein Post-Translational Modifications in Tandem Mass Spectra* **ECML**, 2011, 29(7) 925–932
First-tier publications in Scopus Q2 with main contribution	
7	Beáta Reiz, Michael Myers, Sandor Pongor, Attila Kertész-Farkas* *Precursor Mass Dependent filtering of Mass Spectra for Proteomics Analysis* **Protein and Peptide Letters**, 2014, 21(8) 858–863
First-tier publications in Scopus Q1 journals	
8	Pavel Sulimov, Anastasia Voronkova, Attila Kertész-Farkas* *Annotation of tandem mass spectrometry data using stochastic neural networks in shotgun proteomics* **Bioinformatics**, 2020, 18(5), 2354–2358
9	Uri Keich, Attila Kertész-Farkas, William Stafford Noble *Improved False Discovery Rate Estimation Procedure for Shotgun Proteomics* **Journal of Proteome Research**, 2015, 14(8) 3148–3161
10	Sean McIlwain, Kaipo Tamura, Attila Kertész-Farkas, Charles Grant, ... (+7), William Stafford Noble *Crux: rapid open source protein tandem mass spectrometry analysis* **Journal of Proteome Research**, 2014, 13(10):4488–4491
11	Beáta Reiz, Attila Kertész-Farkas, Michael Myers, Sandor Pongor *Chemical rule-based filtering of MS/MS spectra* **Bioinformatics**, 2013, 29(7) 925–932
Second-tier publications in Scopus Q3-Q4 journals with main contribution	
12	Pavel Sulimov, Elena Sukmanova, Roman Chereshnev, Attila Kertész-Farkas* *Guided Layer-wise Learning for Deep Models using Side Information* **Communications in Computer and Information Science**, 2020, 1086: 50–61
13	Attila Kertész-Farkas, Beáta Reiz, Michael Myers, Sándor Pongor *Database searching in mass spectrometry based proteomics* **Current Bioinformatics**, 2012, 7(2) 221–230
Second-tier publications in Scopus Q4 journals	
14	Beáta Reiz, Attila Kertész-Farkas, Michael Myers, Sándor Pongor *Data preprocessing and filtering in mass spectrometry based proteomics* **Current Bioinformatics**, 2012, 7(2) 212–220

*Senior authorship and correspondence author.

Table 2: Other publications

	Authors, Title, Venue
First-tier publications in Scopus Q1 journals with main contribution	
15	Gleb Filatov, Bruno B. Bauwens, Attila Kertész-Farkas* *LZW-Kernel: fast kernel utilizing variable length code blocks from LZW compressors for protein sequence classification* **Bioinformatics**, 2018, 34(19), 3281–3288
16	Bruna Marini, Attila Kertész-Farkas[1], Hashim Ali,... (+8), Mauro Giacca *Nuclear architecture dictates HIV-1 integration site selection* **Nature** , 2015, 521, 227–231
17	János Juhász‡, Attila Kertész-Farkas‡, Dóra Szabó, and Sándor Pongor: *Emergence of Collective Territorial Defense in Bacterial Communities: Horizontal Gene Transfer Can Stabilize Microbiomes* **PLoS One**, 2014, 9(4)
18	Emily Doughty‡, Attila Kertész-Farkas‡, ... (+4), Maricel G. Kann *Toward an automatic method for extracting cancer- and other disease-related point mutations from the biomedical literature* **Bioinformatics**, 2011, 27(3) 408–415
19	József Dombi, Attila Kertész-Farkas* *Applying Fuzzy Technologies to Equivalence Learning in Protein Classification* **Journal of Computational Biology**, 2009, 16(4) 611–623
20	András Kocsor, Attila Kertész-Farkas, László Kaján, and Sándor Pongor *Application of compression-based distance measures to protein sequence-classification: a methodological study* **Bioinformatics**, 2006, 22(4) 407-412
21	János Z. Kelemen, Attila Kertész-Farkas, András Kocsor, László G. Puskás *Kalman Filtering for Disease-State Estimation from Microarray Data* **Bioinformatics**, 2006, 22(24) 3047–3053
First-tier publications in Scopus Q1 journals	
22	Roman Chereshnev, Attila Kertész-Farkas* *GaIn: Human Gait Inference for Lower Limbic Prostheses for Patients Suffering from Double Trans-Femoral Amputation* **Sensors**, 2018, 18(12), 4146
23	Marina L. Mokrishcheva, Attila Kertész-Farkas, and Dmitri V. Nikitin *New bifunctional restriction-modification enzyme AloI isoschizomer (PcoI): bioinformatics analysis, purification and activity confirmation* **Gene**, 2018, 660, 6-12
24	Roberto Vera,... (+3) Attila Kertész-Farkas, Sándor Pongor *JBioWH: an open-source Java framework for bioinformatics data integration* **Database**, 2013
25	Paolo Sonego, Mircea Pacurar, Somdutta Dhir, Attila Kertész-Farkas, ... (+3), Sándor Pongor *A Protein Classification Benchmark collection for Machine Learning* **Nucleic Acids Research**, 2006, 35, 232–236,
26	László Kaján, Attila Kertész-Farkas, Dino Franklin, Nelly Ivanova, András Kocsor, Sándor Pongor *Application of a simple log likelihood ratio approximant to protein sequence classification* **Bioinformatics**, 2006, 22(23) 2865-2869
Second-tier publications in Scopus Q3-Q4 journals with main contribution	
27	Andrey Shestakov, Danila Doroshin, Dmitri Shmelkin, Attila Kertész-Farkas* *Lookup Lateration: Non-linear Received Signal Strength to Distance Mapping for Non-Line-of-Sight Geo-localization in Outdoor Urban Areas* **LNCS**, 2018, 11179, 234–246
28	Roman Chereshnev, Attila Kertész-Farkas* *HuGaDB: Database for Human Gait Analysis from Wearable Inertial Sensor Networks* **Lecture Notes in Computer Science, Revised Selected Papers**, 2017, 124–134
29	Attila Kertész-Farkas, Somdutta Dhir, ... (+7), Sándor Pongor *Benchmarking protein classification algorithms via supervised cross-validation* **Journal of Biochemical and Biophysical Methods**[2], 2008, 70(6), 1215–1223

Table 3: Other publications (*cont.*)

	Authors, Title, Venue
Second-tier publications in Scopus Q3-Q4 journals with main contribution (*cont.*)	
30	Attila Kertész-Farkas, András Kocsor, Sándor Pongor *Equivalence Learning in Protein Classification* in Machine Learning and Data Mining in Pattern Recognition **Lecture Notes in Artificial Intelligence**, 2007, 4571 824-837
31	Attila Kertész-Farkas, András Kocsor *Kernel-based Classification of Tissues using Feature Weightings* **Applied Ecology and Environmental Research**, 2006, 4(2) 63–71
32	Attila Kertész-Farkas, Zoltán Fülöp, András Kocsor *Compact Representation of Hungarian Corpora* in Hungarian **Hungarian Journal of Applied Linguistics**, 2005, (1-2) 63–70
Second-tier publications in Scopus Q3-Q4 journals	
33	Roman Chereshnev, Attila Kertész-Farkas* *RapidHARe: An Energy-Efficient Method for Real-Time Human Activity Recognition from Wearable Sensors* **Journal of Ambient Intelligence and Smart Environments**, 2018, 10(5), 377–391
34	Dóra Bihary, Attila Kertész-Farkas, ... (+4), Sándor Pongor *Simulation of communication and cooperation in multispecies bacterial communities with an agent based model* **Scalable Computing: Practice and Experience**, 2012, 13(1) 21–28
35	Somdutta Dhir, Attila Kertész-Farkas, ... (+5), Sándor Pongor *Detecting atypical examples of known domain types by sequence similarity searching: The SBASE domain library approach* **Current Protein Peptide Science**, 2010, 11(7) 538-549
36	Róbert Busa-Fekete, Attila Kertész-Farkas, András Kocsor, Sádor Pongor *Balanced ROC analysis (BAROC) protocol for the evaluation of protein similarities* **Journal of Biochemical and Biophysical Methods**[2],2008, 70(6) 1210-1214

‡Equal contribution. [1]Main contribution to the bioinformatics analysis. [2]This journal has evolved into the *Journal of Proteomics* and has continued under a new title and management.

2 Computational Mass Spectrometry

Mass spectrometry is the de facto method to identify molecules in a mixture of samples in several disciplines including proteomics, molecular biology, forensic, pharmaceutical industry, medicine, etcetera. The data obtained from the mass spectrometer undergoes a subsequent computational analysis of a long pipeline of various algorithms, including: (a) spectrum normalization, (b) filtering, (c) pre-processing, (d) in silico generation of candidate peptides, (e) spectrum-peptide-scoring, (f) statistical validation, (g) post-processing, (h) re-scoring, (i) protein assembly, and (j) result validation. This pipeline is illustrated in Figure 2. Each stage has its own standard, imperative programs based on mathematical models. The most commonly used software tools for spectrum annotation and spectrum data analysis are: SEQUEST [3], Mascot [7], X!Tandem [8], Crux [9], Comet [10], Andromeda [11] and MaxQuant [12], MS Amanda [13], Morpheus [14].

In a typical computational analysis, tens or hundreds of parameters are to be specified by the experimenters manually. For instance, CRUX, a software toolkit for mass spec data analysis, has 452, X!Tandem, another software for mass spec data analysis, has 68 multi-choice parameters to be specified by the experimenter. Some of the parameters are straightforward to specify such as the type of the spectrometer instrument used or the information about the organisms examined; other times, it is based on the intuition of the experimenter. For instance, the experimenter has to decide 1) which ions to use in scoring (a,b,c,x,y, or z fragmentation ions), 2) whether the program should consider fully-tryptic or semi-tryptic peptides as well as 3) which amino acid mutations, post-translational and/or chemical-modifications should be considered during the annotation. This is a sort of chicken-egg problem, because these programs are employed to tell us about these deviations.

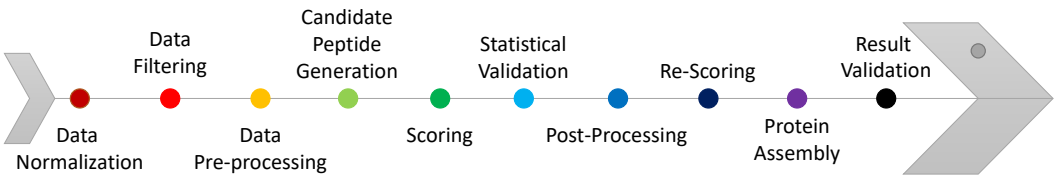

Figure 2: The pipeline of the database-searching-based spectrum annotation.

2.1 Spectrum data generation

This section presents a simplified description of mass spectra for readers not familiar with the field. Of the various and continuously changing techniques of mass spectrometry, we selected the most widely used method, collision induced dissociation (CID), to demonstrate the principle of spectrum analysis. First, a spectrometer selects molecules having the same precursor ion mass-to-charge value and induces a collision of the ions with some noble gas in its chamber. The gas atoms break the peptide ions resulting in fragment ions. The fragmentation sites along with their

names are shown in Figure 3. The a, b, c-ions are associated to the fragment ions containing the N-terminus, and the x, y, z-ions associated to the C-terminus fragment. These fragment types are the most common fragments observed with ion trap, triple quadrupole, and q-TOF mass spectrometers and also called primary fragmentation ions. The most common peptide fragments observed in low energy collisions are b- and y-ions. The secondary fragmentation ion products (SFIPs) originate from primary fragmentation ions by usually losing a mass of a water molecule (H2O) or ammonia molecule (NH3).

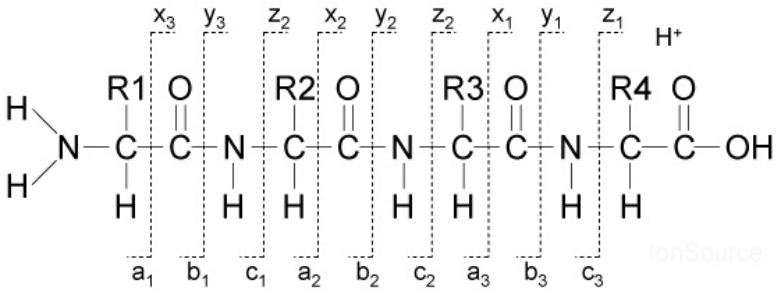

Figure 3: Illustration of the fragmentation sites and the primary ion types. The sites $R1, \ldots R4$ show place of the amino acid residues.

The spectrometer detects the mass-to-charge ratio (x-axis) and measures the abundance (y-axis) of the fragment ions. In principle, the peaks of subsequent b-ions (resp. y-ions) are separated by the mass of an amino acid. For instance, the b_1 and b_2 and the y_1 and y_2 ions on Figure 4 are separated by the mass of the amino acid Q along the x-axis; that is, $b_2 - b_1 = y_2 - y_1 = 128.05858Da$, the mass of the amino acid Q. Therefore, by reading out the amino acids along with the b-ion series (shown by purple lines on Figure 4) provides the peptide sequence from N to C terminal, while along with the y-ion series (shown by blue lines on Figure 4) provides the peptide sequence from C to N terminal. In practice, it is not straightforward how to distinguish between the types of the fragmentation ions in an experimental spectra.

For any given peptide sequence, a so-called theoretical spectrum can be calculated straightforward using the known masses of the amino acids and other few chemical-physical rules. The theoretical spectrum can indicate the types of the primary and the secondary fragmentation ions; however, the prediction of the peak intensities are more complicated, and they all are set to a constant value, usually to unit (1.0) by default. A theoretical spectrum can be considered as an idealized version of the observed experimental spectrum, that is, what one could expect under ideal circumstances. This is in contrast to the experimental spectrum which can be considered a flawed spectrum which is (a) containing many unexplainable peaks, which stem from the unusual fragmentation of the peptide or contaminating molecules, and (b) lacking of expected fragmentation ions, which fail to be observed in the mass spectrometer [15].

2.2 Peptide spectrum scoring

Score functions are the work horses in spectrum annotation methods. Each experimental spectrum is iteratively scored against the set of candidate peptide sequences. The peptide sequences are

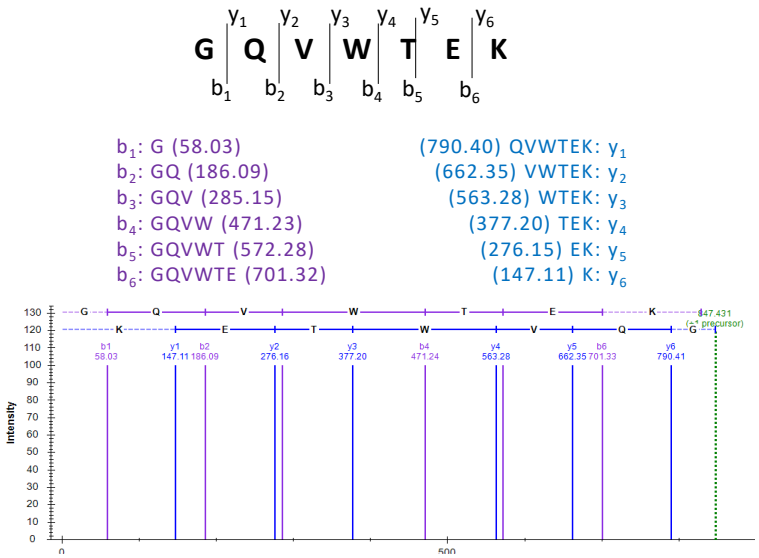

Figure 4: Illustration of the fragmentation of peptide GQVWTEK. The mass of the single charged precursor ion is 847.43 Dalton, which can be calculated by summing up the mass of the amino acids, the terminal residues (H and OH) and the mass of one proton (~ 1 Da). The numbers in the paranthesis for each fragmentation ion show the calculated mass of the ions.

first converted into the so called theoretical spectrum calculated as described above. The scoring of a theoretical and experimental spectra typically provides a similarity-like score, i.e. higher score indicates a better match, based on matching the peaks in the two spectra. Good score functions are:

1. *discriminative*, meaning they separate correct peptide-spectrum-matches (PSMs) from the incorrect ones,
2. *well-calibrated*, meaning they have a well defined and accurate semantics,
3. *unbiased*, meaning that they assign each spectrum to incorrect target or decoy peptide with equal likelihoods,
4. *universal*, meaning they work well for spectra generated using diverse configurations of MS instruments and experimental protocols [16, 17].

In this section, first we introduce the most known scoring methods and spectrum preprocessing steps, and the we discuss their properties.

2.2.1 Spectrum discretization

Because of the little imprecision of the detector of the mass spectrometer, the location and the intensity of the peaks in the experimental spectrum are inexact and they would not match to the exact peaks in the theoretical spectrum. In order to handle the imprecision, a small tolerance is introduced. In practice, this is mostly done by dividing the observed spectra along the m/z into small discrete bins resulting in a real-valued vector v, whose components contain the sum or the maximum of the intensities of the peaks which fall in that particular bin. Formally, a peak at position p (m/z) is placed in the bin $k = \left[\frac{p}{w} + o\right]$, and the value of the $v[k]$ is the sum

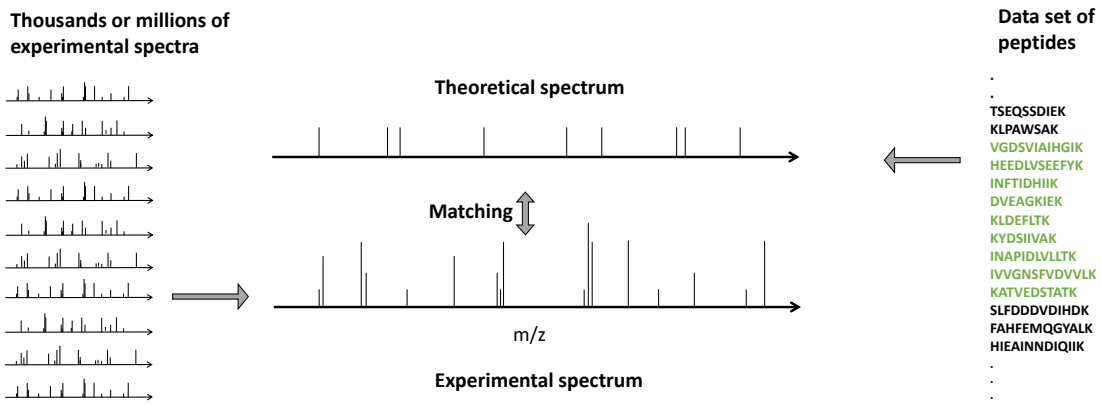

Figure 5: Illustration of the database-searching-based peptide spectrum matching. Every experimental spectrum is matched to every candidate peptide. The data set of peptides is illustrated on the RHH, the candidate peptides are marked with green. The set of candidate peptides vary upon the experimental spectrum. The peptide sequence is first converted to a theoretical peptide with uniform peak intensities. The experimental spectrum is shown at the bottom with different peak intensities. The matching of an experimental and a theoretical spectra is evaluated with score functions. Experimental spectrum is annotated with the best scoring candidate peptide sequence.

or the maximum of the intensities of the peaks falling in bin k. Here, [.] denotes the standard rounding operator. For modern spectrometers with detectors of high resolution, the bin-width w is usually set to 0.05 and the offset $o = 1.0$ referred to as high-resolution fragmentation settings (HRFS), while for detectors of low-resolution capacity the bin-width $w = 1.0005079$ and the offset $o = 0.6$ referred to as low-resolution fragmentation settings (LRFS). Therefore, if the heaviest observable fragment ion is of 2000 m/z then the discretization step results in a vector with 40,000 bins (or also called a 40,000-dimensional vector) for HRFS, and a vector with 1999 bins for LRFS. We note that, the resolution setting must be specified with respect to the spectrometer. The vector v resulted is called the discretized spectrum. The spectrum discretization is done for the experimental and for the theoretical spectra respectively but with the same parameters.

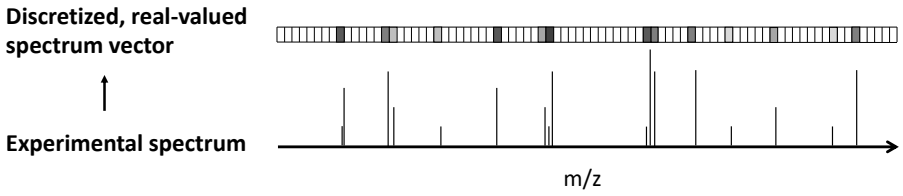

Figure 6: Illustration of the spectrum discretization. The bottom represents an experimental spectrum. The vector above represents the corresponding discretized spectrum vector. Vector bins without peaks are colored with white. More intense peaks fall in a vector bin, the darker color it shaded with.

In the rest of this thesis, any spectrum s_i or h_j indicates a discretized spectrum vector.

2.2.2 Spectrum filtering and preprocessing

Spectra often preprocessed and filtered beforehand scoring. The most common steps are:

- *Noise filter.* The filter removes low intensity peaks, often called grass peaks, whose intensity is below the q% of the most intense peak. For instance, SEQUEST defines $q = 1$% while

OMSSA system uses $q = 2.5\%$.

- *Top-N filter.* This methods retrains the N most intense peaks in the experimental spectrum and disregards all the others.
- *Top N in Y regions filter.* It was designed to overcome the issue that high intensity peaks can often cluster in certain parts of a spectrum. With this approach, the spectrum is divided into Y equal regions (defined with a certain overlap), and the top N intensities are retained in each of the regions [18].
- *Top-N intensity in a window of Z filter.* This approach first sorts the peaks by decreasing intensity, then goes from the first peak till the last and in each iteration retain the top N intensities in a window of $\pm Z$ m/z right and left of the most intensive peak and exclude all other peaks in the window [19]. The value of Z is usually set to be smaller than the mass of an amino acid, because true peaks are not expected to occur with spacing less than that of an amino acid. Z is usually set to 1 or 2 to account for ions from two different ion series occurring in the same small window [20].
- *Removal of the precursor ion-related peaks.* This step deletes all peak in a small vicinity, usually $\pm 1 - 3$ Dalton, of the precursor ion.

Spectrum annotation systems may use any of the methods above in any combination. For instance, the SEQUEST, Comet, and Tide program first removes all peaks around the precursor ion in a window of 1.5 Da (by default), then it replaces the peak intensities by their square root, removes the small peaks which are less than the 1% of the most intense peak, and finally spectra are divided into 10 equal-length regions along the (m/z) axis and intensities in each segment are scaled separately so that the intensity of the highest peak in each segment is 50. The X!Tandem employ the TOP-N filter with N=50 by default.

2.2.3 Score functions

Score functions are essentially based on matching experimental peaks to theoretical peaks. This is illustrated in Figure 7. Standard score functions are:

- Shared Peak Count (SPC) defined as

$$\mathrm{SPC}(s_i, h_j) = \sum_{k=1}^{N} \mathbb{I}(s_i[k] \neq 0) \times \mathbb{I}(h_j[k] \neq 0), \tag{2}$$

  where N is the number of bins. This approach counts the peaks at common locations in both in the theoretical and the experimental spectra, but does not take peak intensities into account.

- Inner product (IP):

$$\mathrm{IP}(s_i, h_j) = s_i^T h_j. \tag{3}$$

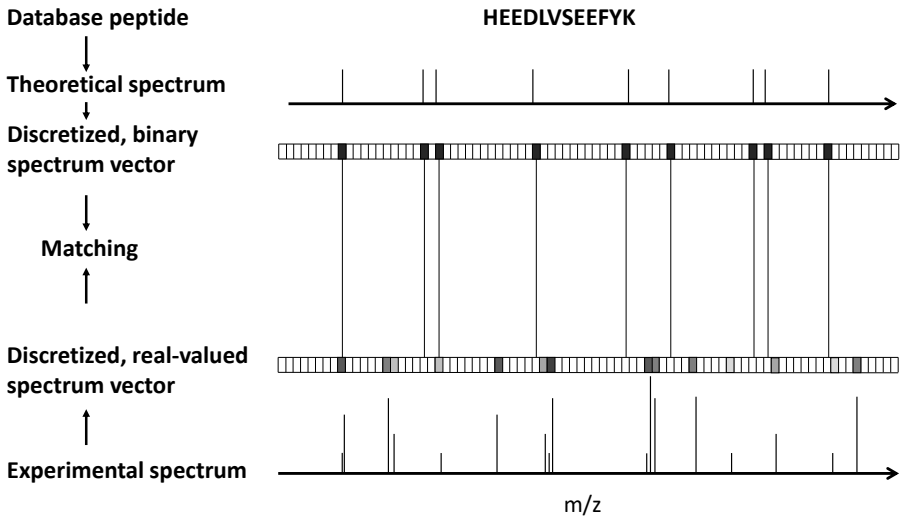

Figure 7: Illustration of the PSM scoring. In this example, there are two matching peaks, i.e. $SPC = 2$. If the intensity of the theoretical peaks is 1.0 and the experimental peak intensity is scaled between 0 and 1, then $IP \approx 1.1$. This does not seem to be a good match.

This function takes the intensities of the matching peaks into account. Note that if the intensities of the theoretical peaks are set to 1, then the discretized theoretical spectrum vector is essentially a binary vector, and the IP measures the sum of the intensities of the matching peaks of the experimental spectra.

- HyperScore, introduced by Fenyo et al. [8]:

$$\text{HyperScore}(s_i, h_j) = \text{IP}(s_i, h_j) \times N_b! \times N_y! \tag{4}$$

where $N_b!$ is the factorial of number of matched b-ions and $N_y!$ is the same but for matched y-ions of the theoretical spectrum. The idea behind is that, for example, 4 matching theoretical peaks from the b-ions series (or 4 matching peaks from the y-ion series) may indicate better spectrum peptide match than matching 2 b-ions and 2 y-ions, which looks more like a random match.

- Cross correlation scoring (XCorr) was introduced with SEQUEST [3] as

$$\text{XCorr}(s_i, h_j) = \text{IP}(s_i, h_j) - \frac{1}{151} \sum_{\tau=-75}^{+75} \text{IP}(s_i, h_j[\tau])), \tag{5}$$

The first part simply qualifies the match between the experimental and theoretical spectra using the inner product of the corresponding vectors. The second part, also referred to as cross-correlation penalty, provides an estimation of the mean of the null distribution from 151 random matches obtained with a random theoretical peptide $h_j[\tau]$ generated by shifting the components of vector h_j by τ steps. We note that theoretical spectra correspond to real

peptide sequences, while shifting its components by $\pm\tau > 0$ steps breaks its semantics, and it cannot be associated with any real peptide sequence of the original mass, hence resulting in a random vector. Consequently, the XCorr score returns the signed difference between the match score and an estimated mean of the null distribution. When s_i and h_j are unrelated then $\mathrm{IP}(s_i, h_j) \sim \frac{1}{151}\sum_{\tau=-75}^{+75}\mathrm{IP}(s_i, h_j[\tau]))$ and $\mathrm{XCorr}(s_i, h_j) \approx 0$; however, when s_i and h_j are indeed related then $\mathrm{IP}(s_i, h_j) \gg \frac{1}{151}\sum_{\tau=-75}^{+75}\mathrm{IP}(s_i, h_j[\tau]))$ and $\mathrm{XCorr}(s_i, h_j) \gg 0$.

- The score function employed in the Andromeda system [11] is defined as:

$$\mathrm{A}(s_i, h_j) = \max_{q,loss}\left\{-10\log\sum_{j=k}^{n}\left[\binom{n}{j}\left(\frac{q}{100}\right)^{j}\left(1-\frac{q}{100}\right)^{n-j}\right]\right\}, \tag{6}$$

  where n denotes the total number of theoretical peaks, k denotes the number of matching peaks, i.e. $k = SPC(s_i, h_j)$, q indicates the number of the most intense peaks in every 100 Dalton. Finally, *loss* is a boolean option whether or not to consider secondary fragmentation ion products (SFIP) during scoring. Therefore, the scoring runs twice, once with and once without considering SFIPs. Note that, the scoring here is essentially the SPC with using an optimization whether to consider SFIPs or not and on the number q of the highest intensity peaks that are taken into account per 100 m/z interval. Also notice that the *sum* part calculates the statistical significance (p value) of having k or more randomly matched peaks. This lies on the assumption on that peaks in spectra are distributed uniformly.

- The score function employed in the retired OMSSA system is defined as the p-value of the number of the matches SPC, assuming that the null distribution follows a Poisson distribution, formally:

$$\mathrm{O}(s_i, h_j) = \frac{\mu^k}{k!}e^{-\mu}, \tag{7}$$

  where $k = SPC(s_i, h_j)$ and μ is the mean of the Poisson distribution.The mean is estimated as follows:

$$\mu(s_i, h_j) = \frac{2t}{r-o}\frac{h(r-o)}{m} = \frac{2tvh}{m}, \tag{8}$$

  t denotes the matching tolerance, the experimental spectrum has v different peaks, and the lightest peak is at o and the heaviest peak is at r (m/z) in the experimental spectrum, and m denotes the mass of the precursor. Then a measure of the number of the possible matches is $(r-o)/2t$. Finally, h denotes the number of the theoretical peaks.

Scoring many unrelated peptides against a single experimental spectrum would yield a null-distribution. This null-distribution is different for every spectra, hence we refer to it as spectrum-specific null-distribution. In the next section, we discuss methods how to separate the score of the correct annotation from the spectrum-specific null distribution and after that we discuss methods how to normalize the top-score relative to the spectrum-specific null-distribution.

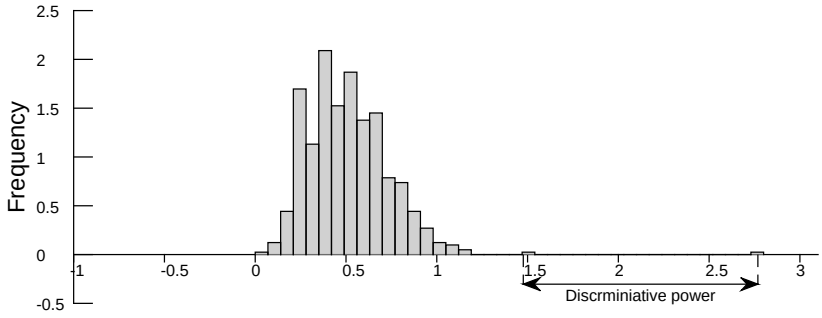

Figure 8: Illustration of the discriminative power using the score distribution of the spectrum with ID=9592 from the Malaria dataset. The score associated to the best-scoring peptide is around 2.75. The discriminative power intuitively tells how well a PSM score of the best-scoring peptide is separated from the null-distribution.

2.3 Discriminative property

The discriminative ability of a scoring functions means that the scores of a correct PSM is much higher, and therefore it is well separated from the score distribution of the incorrect PSMs (i.e. the null distribution); therefore, the correct PSMs can be separated from incorrect ones using a simple threshold.

Unfortunately, the score functions in spectrum identification are hindered by (a) the presence of many unexplained peaks, which stem from the unusual fragmentation of the peptide or contaminating molecules, or (b) the lack of expected fragmentation ions, which fail to be observed in the mass spectrometer [15]. Score functions attempt to mitigate the negative effects caused by these issues (a) by considering secondary fragmentation ion products (SFIP), such as the ions derived from water, carbon monoxide, or ammonia losses, in addition to primary fragmentation ions. For instance, Andromeda [11] generates auxiliary peaks for water or ammonia loss products for theoretical peptides containing D, E, S, T or K, N, Q, R amino acids, respectively. The SEQUEST system [3, 21] additionally incorporates signals from the flanking bins of the discretized spectrum vector [22], SFIP, and highly charged theoretical fragmentation ion masses depending on the charge state of the precursor ion. The XCorr puts a weight of 50 on the matching primary fragmentation ions, usually b and y ions, a weight of 25 on the matching flanking peaks, and a weight of 10 on the matching peaks of SFIP [3, 5].

This can essentially be regarded as a weighted sliding window technique represented with a vector. For LRFS, the vector may consist of 61 bins, in which there is a weight of 50 in the center of the weight vector (at the 31st bin), weights of 25 beside the center for flanking bins (the 30th and 32nd bins), and weights of 10 for the SFIPs that are 17,18, 28 steps towards the lower end (the 14th, 13th and 3rd bins) from the center, while all other elements in the weight vector are zeros. Such sliding windows can augment a peak by the weighted sum of the intensities of nearby peaks if peaks in a small range match to the pattern defined by the sliding window. Figure 9 illustrates the matching of an experimental and a theoretical spectrum by considering SFIP using such a sliding window technique.

For further increase the discriminative power, the XCorr scoring methods additionally generate the theoretical peaks with $n-1$ charge states (or up to a user defied limit) for experimental spectra

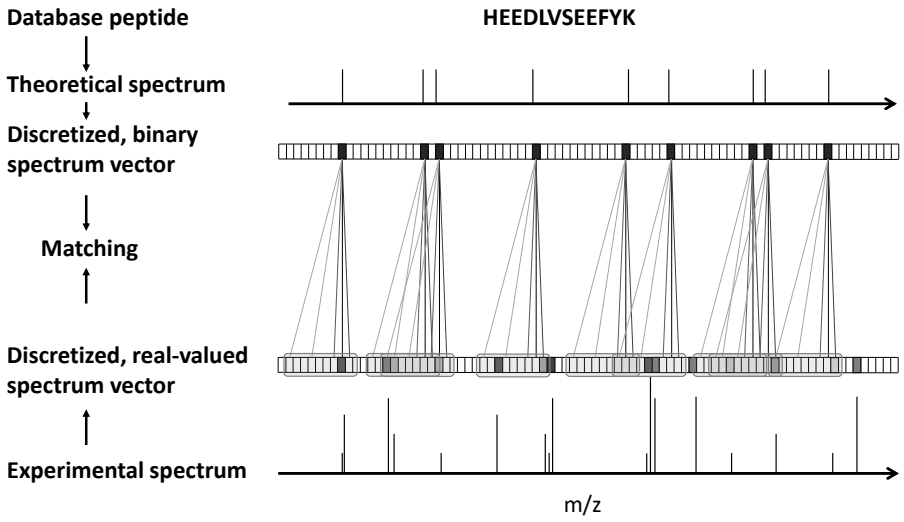

Figure 9: Illustration of the scoring considering SFIPs.

with precursor charge state $n > 2$.

OMSSA also employs an additional rule in order to increase its discriminative power. It requires that at least one of the theoretical peaks must match to any of the experimental peaks among the top n ($n = 3$ by default) intensive peaks. This requirement changes the null distribution model and it becomes:

$$\mathrm{O}'(s_i, h_j) = \frac{1}{Q}(1 - (1 - q)^x)\mathrm{O}(s_i, h_j), \tag{9}$$

where q is the probability of that an experimental peak matches to a theoretical peak is $q = n/v$, and Q is a normalization factor defined as $Q = \sum_x (1 - (1 - q)^x)\mathrm{O}(s_i, h_j)$.

2.4 Score calibration

Uncalibrated raw PSM scores may indicate different match quality for different spectra. For instance, the distributions of the top scoring PSMs of doubly and triply charged spectra shown in Figure 11 indicate that a raw score of 2.5 may imply a correct annotation for a doubly but an incorrect annotation for a triply charged peptide molecule [4]. Spectrum-specific score calibration methods aim to provide a sort of score normalization so that spectrum assignments become comparable with each other; therefore, a single threshold can be selected to accept or reject spectrum annotations. The calibration allows one to obtain many more spectrum annotations at any desired false discovery rate (FDR) [4]. Score calibration methods involve a null distribution and calibrate a raw score to either the mean or the tail of the null distribution.

The standard approach of score calibration is to assign a spectrum-specific statistical significance to a raw PSM score by estimating a probability of observing a random score equal to or greater than the observed PSM score. This is the p-value, which in fact has well-defined and accurate semantics [23, 24, 25] over various experimental protocols and diverse configurations of MS instruments. The success of the score calibration methods relies on how well they approxi-

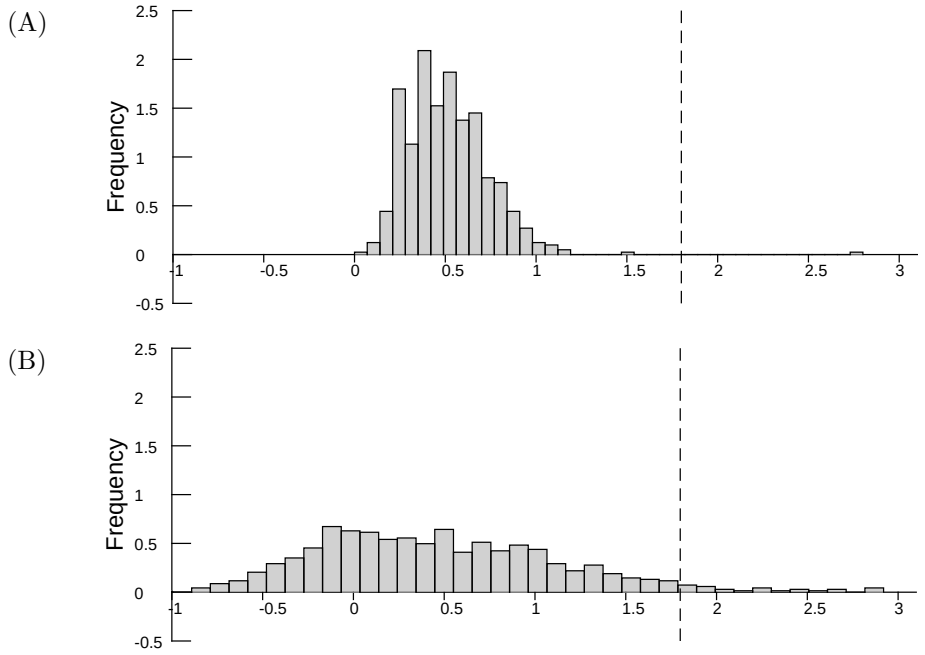

Figure 10: Spectrum specific null distributions of two different spectra from the Malaria dataset. (A) The null distribution of a spectrum with scan ID=9592 and a charge state of 2. (B) The null distribution of a spectrum with scan ID=13963 and a charge state of 3. The decision threshold 1.8 would be appropriate for the spectrum on top, but it would be inappropriate for the second spectrum.

mate the tail or the extreme tail of the null distribution to obtain a p-value estimation. There are various approaches on how to assign a statistical p-value to a PSM score. Figure 12 illustrates and compares the principles of the score calibration methods on a null distribution. Below, we discuss each approach in detail.

2.4.1 Analytical approaches

Some methods employ analytical models, such as a binomial distribution in Andromeda [11] and MS Amanda [13], Poisson distribution by Open Mass Spectrometry Search Algorithm (OMSSA) [20], a Weibull distribution for the XCorr [26], or a Gumbel distribution for Spectrum Specific P-value (SSPV) [27], and rely on the assumption that peaks match independently between spectra. The disadvantages of these models include that (a) this assumption is not justified in practice [28] and (b) the analytical probability mass functions (PMF) of binomial or Poisson distributions do not have cumulative distribution functions in closed forms to calculate the p-value instantly. As a result, they require a longer CPU time to sum over a larger number of PFMs at hypothetical PSM scores. The parameters of the exponential distributions (Weibull and Gumbel) are fitted from empirical PSM scores separately for each spectrum.

Let us demonstrate an analytical approach with an example for the SPC score function from [29]. Let N denote the total number of theoretical peaks from all the candidate peptide sequences from the database for a given spectrum s and let K denote the total number of theoretical peaks which matched to any of the experimental peaks from s_i, ($K \leq N$). That is, if s_i has, say, 5 candidate peptides $h1, \ldots, h_5$, then N is the sum of the theoretical peaks of all the 5 candidate

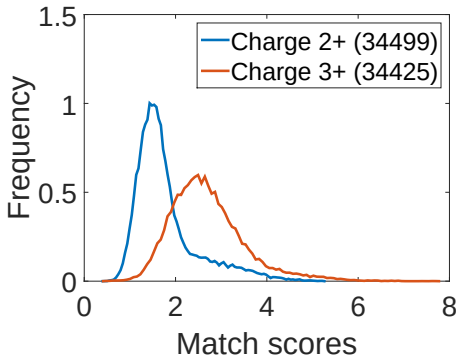

Figure 11: Distributions of top scoring PSMs obtained with simple match on Yeast data for doubly (blue) and triply (red) charged precursor ions.

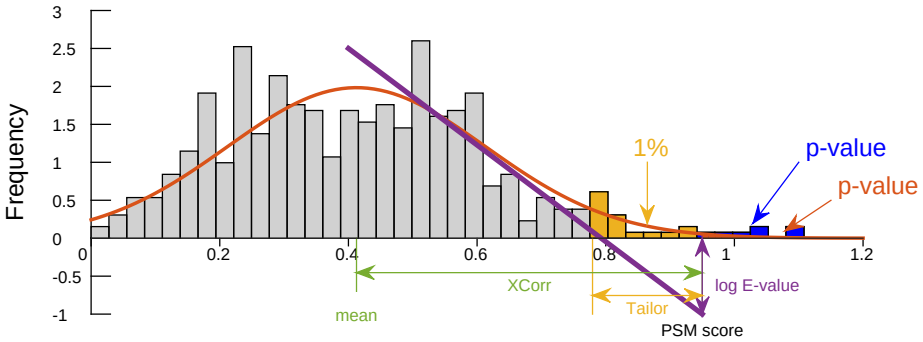

Figure 12: Illustration of the principles of the PSM score calibration approaches on a null distribution denoted by grey. The null distribution was obtained during scoring a real spectrum. (Green) XCorr calibrates the matching score by measuring the difference between the PSM score and an approximation of the mean of the random matching scores (Comet, Sequest, Tide). (Purple) Regression based methods fit a liner line on the empirical survival function based on the histogram of the random scores and extrapolates an E-value where a PSM score falls on this regression line (Comet, X!Tandem). (Blue) empirical p-values are calculated from the exact null distribution obtained with dynamic programming methods (XCorr exact p-value in Tide, MS-GF+) or with Monte-Carlo techniques [4]. (Red) P-values are calculated by using analytical probability density functions (OMSSA, Andromeda, Morpheus, SSPV, Weibull calibration of XCorrs). (Yellow) Tailor methods calibrates the score to the top 100-quantiles, i.e. relatively to the score which has a p-value of 0.01.

peptides, and K is the total number of the matching peaks produced by the 5 candidate peptides. Now, let us consider only one candidate peptide h_j witch has N_1 theoretical peaks and among those $K_1 \leq N_1$ peaks are matched to any of the peaks of the experimental spectrum s_i. The probability of that we observe $\text{SPC}(s_i, h_j) = K_1$ matching peaks among N_1 theoretical peaks when we match a peptide h_j can be modeled with a hypergeometric distribution and it can be calculated by

$$P_{K,N}(K_1, N_1) = \frac{\binom{K}{K_1} \cdot \binom{N-K}{N_1-K_1}}{\binom{N}{N_1}} \tag{10}$$

Using Eq. 10, a statistical p-value of a PSM producing $\text{SPC}(s_i, h_j)$ or higher matches randomly can be

$$\text{P-val}(\text{SPC}(s_i, h_j) = X) = \sum_{i=X}^{N_1} P_{K,N}(i, N_1) \tag{11}$$

Notice that, this model seems to be affected by the number of the candidate peptides. If there is exactly one theoretical peptide in the candidate peptide set CP, therefore $N = N_1$ and $K = K_1$, this approach would produce $P_{K,N}(K_1, N_1) = 1$.

2.4.2 Linear approaches

X!Tandem [8] and Comet [10] fits a linear regression line to the estimated survival function of the null distribution to calibrate the score for each experimental spectrum. Comet employs a log transformation of the survival function, and fits a linear regression line, and calculates a calibrated score, an E-value, by extrapolating the linear regression model at the top-scoring PSM score. X!Tandem employs a similar approach; it fits a linear regression line to the empirical survival function of the log of the HyperScores [8]. Both approaches assume that the tails of the null distribution decays exponentially; however, this assumption has not been critically analyzed.

The drawbacks of score calibration methods based on fitting specific parametric models includes that they cannot be straightforwardly generalized to other score functions and that the parametric distribution whose parameters are estimated using the overall distributions of PSM scores might not be accurate at the extreme tail [27].

2.4.3 Exact approaches

Another type of p-value estimation methods exploits the exact null distribution obtained from scoring all possible peptide sequences against the query spectrum s_i, which have the same − up to an instrument specific tolerance − precursor mass as s_i [24, 25, 17, 30, 31]. The explicit enumeration of all peptide sequences is computationally unfeasible; however, their scoring would involve re-calculating many (in fact, exponentially many) redundant "sub-scoring". For instance, consider two peptide sequences: $h_1 = TGHLLW$ and $h_2 = TGHLLA$, which share a common prefix $p = TGHLL$. The calculation of $c_1 = SPC(s_i, h_1)$ and $c_2 = SPC(s_i, h_2)$ would require recalculating the sub-score $c = SPC(s_i, p)$ twice; however, if it was calculated and stored once,

then c_1 and c_2 could be calculated by using c as $c_1 = c + c_W$ and $c_1 = c + c_A$, where c_W and c_A denote the score gain produced by the amino acid W and A appended to p, respectively. This is the idea which exact p-value estimation methods rely on, and they employ dynamic programming approach to store such sub-scores to avoid score re-calculation and obtain exact null distribution in a feasible (polynomial) time.

Let us discuss this method in detail. This method requires the following elements:

1. A query spectrum s_i. The score null distribution will be calculated specifically for the experimental spectrum s_i.
2. Score function. For now, let us use the SPC score function from Eq. 2.
3. Mass discretization $m(.)$. This is the method used to discretize molecular masses. For the sake of simplicity, let us use the integer discretization which keeps the integer part of the mono-isotopic masses, defined as $m(a) = \left[a + 0.5\right]$ for a real valued mass a. For instance, the mass of glycine (G) is 57.021463735, and its discretized mass is $m(G) = 57$, the mass of hydrogen (Hy) is 1.007825035, its discretized mass $m(Hy) = 1$. This discretization must strictly corresponds to the spectrum discretization.
4. This method needs a two-dimensional, dynamic programming table, denoted by D. The columns of D correspond to discretized masses and the rows of D correspond to scores. For a given experimental spectrum s_i, the cell in column c and row r of D (denoted as $D[r, c]$) contains the number of peptides having a discretized mass of c and producing a score of r with spectrum s_i. Therefore, the numbers in the column corresponding to the discretized neutral mass of s_i renders the distribution of scores which would be obtained by scoring the s_i against all the possible peptide sequences. This is illustrated in Figure 13.

The calculation starts with filling the table D with 0s. Next, a "start" score distribution is calculated for the mass of the N-terminal residue (hydrogen or a proton) in the column $c = m(Hy) = 1$. There is only one N-terminal residue and it scores exactly zero with any spectrum s_i because this residue is not generated in the theoretical spectra. Thus, $D[0, 1] = 1$ and $D[r, 1] = 0$ for $r > 0$. This distribution is correct, because there are no other residues with a discretized mass of 1 which could score with any experimental spectrum. Then, the score distributions for any discretized masses in dynamic programming table D are calculated iteratively for each column $c > 1$, for each row r, and for each amino acid a by

$$m' = c + m(a) \tag{12}$$

$$s' = r + \mathbb{I}(s_i[m'] > 0) \tag{13}$$

$$D[s', m'] = D[s', m'] + D[r, c], \tag{14}$$

where $m(a)$ is the discretized mass of the amino acid a and $s_i[m']$ is the intensity of the peak at position m' in the experimental spectra s_i. The $s' = r + \mathbb{I}(s_i[m'] > 0)$ means that we increase the score by 1 according to the SPC score function if there is a peak at position m' in s_i.

The idea behind this is the following. Consider an initialized table D and the Figure 13. Now, having the start distribution for the residues with a mass of 1 in $c = 1$, we can calculate a score

distribution for the peptide fragment G, that is a peptide fragment consisting of only the glycine amino acid. The discretized mass of this peptide fragment is $m' = 58$ which is the discretized mass of the N-terminal residue c and mass of the glycine amino acid $m(G) = 57$. This peptide would produce a score $s' = 1$, because there is a peak at position $m' = 58$ in the experimental spectrum s_i as illustrated in Figure 13, therefore $D[1, 58] = 1$ and all other elements in the column 58 are zero. This distribution is correct again, because there is only one residue with a discretized mass of 58 and the corresponding score is exactly 1.

Now, let us consider the score distribution for the discretized mass of $m' = 464$. There are two peptide fragments, in this example, having a mass of 464 and producing a score of 2: "YGGKG" and "YGWA" and there are two another peptide fragments with this mass but producing a score of 3: "GYGKG" and "GYWA". Therefore, $D[2, 464] = 2$ and $D[3, 464] = 2$. Therefore, the column corresponding to the neutral mass of the peptide molecule will contain the null-distribution for the experimental spectrum s_i.

A detailed pseudo code for the exact p-value method using the inner product scoring function defined in Eq. 3 can be found in Algorithm 1.

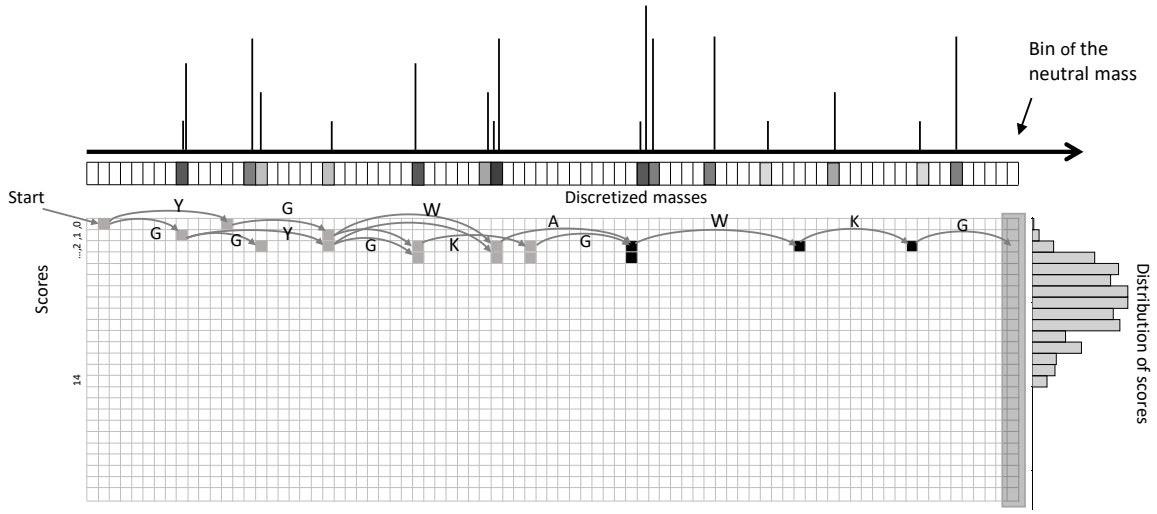

Figure 13: Illustration of the calculation of score distribution by the dynamic programming method. The value in the start at position $D[c, m] = 1$, where c indicates the row corresponding to score of 0, and m is the bin of the N-terminal residue (i.e. the mass of the hydrogen with the mass of N-terminal modification).

The exact methods, indeed, result in a perfect score calibration; however, they have several conditions and drawbacks.

1. The method above cannot handle the ions of the C-terminal (x, y, z-ions) directly in the scoring function. This issue is circumvented in practice by treating every experimental peak p as a b-ion peak and introducing a new peak $p' = NM(s_i) - p$ into the experimental spectrum, where $NM(s_i)$ denotes the neutral mass corresponding to the observed peptide molecule s_i. Therefore, scoring s_i against a theoretical spectrum containing b and y-ions

Algorithm 1: Score count calculation.

Input : v: experimental spectrum vector containing peaks with positive intensity values; A: list of discretized masses of the amino acids; P_N, P_I, and P_C: frequencies of the N-terminal, inner, and C-terminal amino acids (resp.).

Output: The scores in the last column.

$T \leftarrow$ bin of the neutral mass of the spectrum.
$C \leftarrow$ largest discretized score.
$D \leftarrow$ dynamic programming table with size of $C \times T$, initialized with zeros.
$z \leftarrow$ the index of the row corresponding to the score of 0.
$n \leftarrow$ the bin index of mass of N-terminal residue.
$D[z, n] = 1$.
for $a = 0 \rightarrow length(A)$ **do**
  $m = n + A[a]$
  Pass **if** $m > T$
  $s = z + v[m]$
  $D[s, m]$ += $D[c, n] \cdot P_N[a]$
end for
for $c = z + 1 \rightarrow T$ **do**
  for $r = 0 \rightarrow C$ **do**
    Pass **if** $D[r, c] = 0$
    for $a = 0 \rightarrow length(A)$ **do**
      $m = c + A[a]$
      Pass **if** $m > T$
      if $m < T$ **then**
        $s = r + v[m]$
        $D[s, m]$ += $D[r, c] \cdot P_I[a]$
      else
        $D[r, m]$ += $D[r, c] \cdot P_C[a]$
      end if
    end for
  end for
end for

produces the same score as scoring s'_i against a theoretical spectrum containing only b-ions.

2. The null-distribution calculated above makes the assumption that all amino acid sequence are *a priori* equally likely, while the peptide sequences from the CP set would produce a slightly different null distribution and would yield a biased p-value for the elements in CP. This problem can be solved by considering the relative frequencies of the amino acids found in the protein sequence dataset and by including these frequencies to Eq. 14: as

$$D[s', m'] = D[s', m'] \cdot P(a) + D[r, c], \tag{15}$$

where $P(a)$ is the relative frequency of the amino acid a in the protein sequence database.

3. The calculation of the elements of the dynamic programming table requires a significant amount of CPU time.
4. The dynamic programming approach requires the score function to be additive [30]. For instance, the exact p-value approach would not work with HyperScore.
5. The dynamic programming method fails for peak-matching-based score functions (e.g. XCorr) used with data of high-resolution fragment mass accuracy because the fragmentation ions are approximated by the sum of the discretized masses of the amino acids in the dynamic programming method, which in turn can be different from the discretized mass of the whole fragmentation ion in high-resolution settings. That is, for a mass discretization $m(a) = \left[a/w + 0.5\right]$ and for a peptide fragment, say, YGGKG, the equation $m(YGGKG) = m(Y) + m(G) + m(G) + m(K) + m(G)$ fails when $w < 1$. Note that, for low-resolution MS2 data, the $w = 1.0005079$ and the information loss due to discretization hardly poses any problems in practice. This is discussed in details by Lin et al. [31].

To overcome the issues (3-4) mentioned above, empirical p-values of PSMs can be estimated via scoring spectra against a large number, say 10K, of decoy peptide databases [4]. In this scenario, well-calibrated p-values can be obtained for any type of score function using with high- or low-resolution MS2 data, albeit at additional expense of CPU time.

2.4.4 Heuristic approaches

A method, called Tailor method, calibrates PSM scores to the tail of the observed null distribution, where random scores are observed during the database search step, but not to the extreme tail, such as the exact p-value (XPV) methods, where samples are rare.

Tailor works as follows. Let us consider an experimental spectrum s that is matched to N different candidate peptide sequences during the database searching step resulting in the following positive PSM scores: $s_1, s_2, \ldots, s_N > 0$. Let us assume, for now, that N is large enough and that these scores are sorted in decreasing order; thus, the experimental spectrum s is to be annotated with the peptide sequence that produces the score s_1. These scores form the basis of an empirical null distribution for the spectrum s. The 100-quantiles define 99 cut points dividing the range of the score distribution into 100, continuous intervals with equal probabilities. The last (99th)

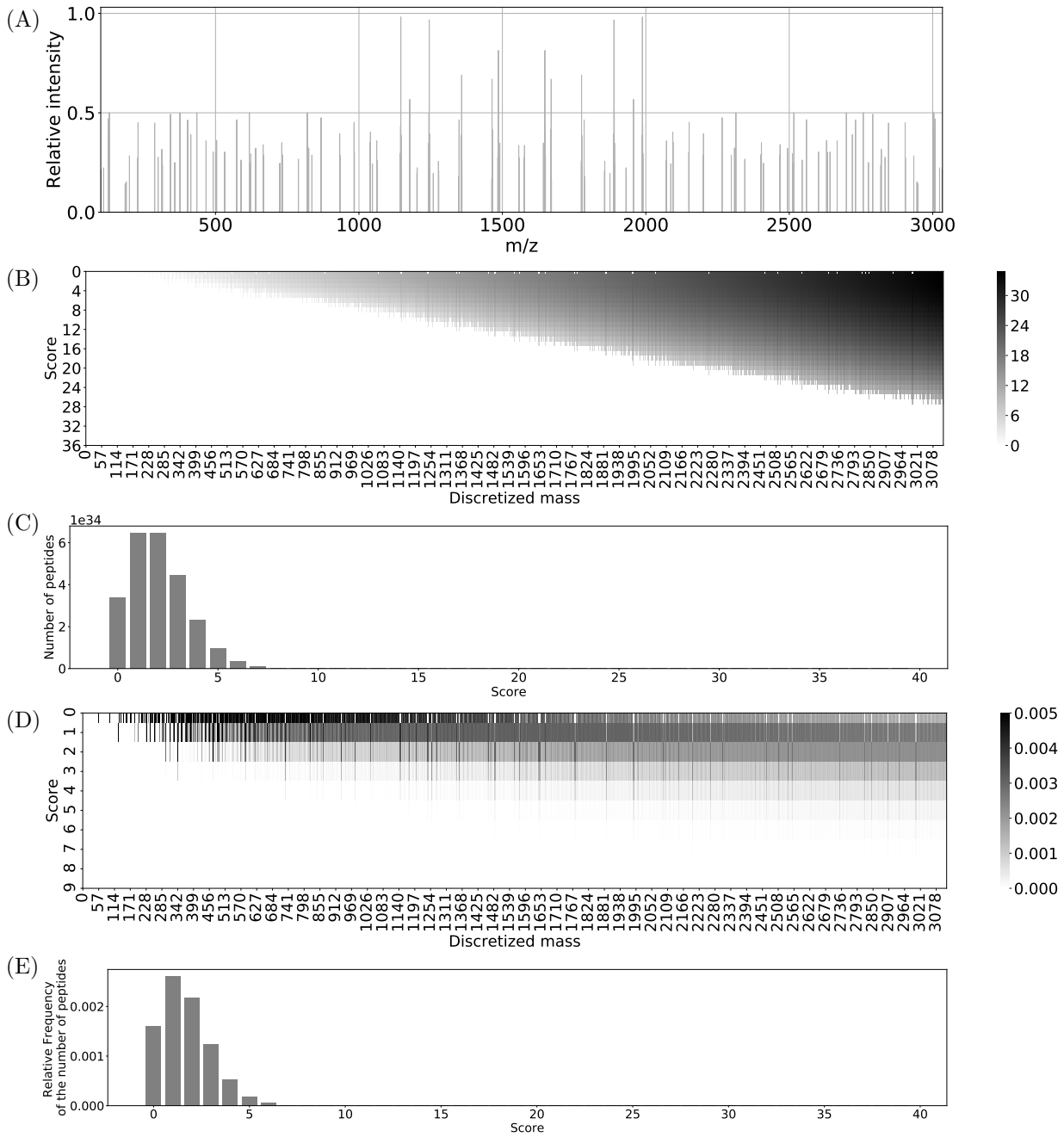

Figure 14: Example of the score statistics in the dynamic programming table for an experimental spectrum. (A) Illustration of the experimental spectrum after including evidence peaks. The experimental spectrum with scan id=13963 and neutral mass=3133.62 Da from the Malaria dataset. (B) The heatmap representation of the dynamic programming table containing the numbers of the scores for the SPC score function. The heat map scale is logarithmic. The dynamic programming table was calculated with the SPC scoring, $m(a) = \left[a/1.0005079 + 0.5\right]$ discretization. (C) The bar plot of the number of peptides from the dynamic programming table of the column corresponding to the neutral mass of the experimental spectrum. (D) The heat map representation of the dynamic programming obtained with the IP scoring, $m(a) = \left[a/1.0005079 + 0.5\right]$ discretization, and using amino acid frequencies as shown in 1. (E) Same as the plot (C), but for the plot (D).

score of the 100-quantiles of the empirical null distribution, denoted by Q100, is obtained here by selecting the PSM score at the position $i^* = [N/100]$, where [.] denotes the standard rounding

operation. Therefore, $Q100 = s_{i^*}$ and the Tailor method calibrates the raw PSMs scores by

$$\tilde{s}_i = \frac{s_i}{Q100} \tag{16}$$

for $i = 1, \ldots, N$.

Tailor is quick and works with any score functions, albeit it is less accurate (because it is a heuristic and a non-parametric approach), whereas exact methods are accurate, albeit slow and require specific score functions.

2.5 P-value calibration

Having discussed several methods for p-value estimation yields to the question, if there are so many of them which method is correct. It is important to keep it in mind that these methods merely provide an estimation on the true, but unknown p-value, and an estimation can be accurate or inaccurate. If a p-value calculation method systematically produces more significant p-values, i.e. smaller numbers, than the true (but unknown) p-values, then it is called a liberal or optimistic; otherwise, if it systematically produces less significant p-values, i.e. larger numbers, than the true p-values, it is called a conservative estimation.

Luckily, there is a way to provide an estimation on the bias in p-values. It is known that, sampling from a distribution, the corresponding p-values are uniformly distributed. In order to show this, let us assume we are given a continuous, cumulative distribution function $F_X(X)$ and it is invertible, i.e. F_X^{-1} exists. We want to show that, the p-values $Z = 1 - F_X(X)$ are uniformly distributed, that is $F_Z(X)$ is uniformly distributed. Note that $Z \in [0, 1]$. For the uniform distribution on $U \sim U_{[0,1]}$, we also have $1 - U \sim U_{[0,1]}$ uniformly distributed, furthermore $P(U \leq u) = u$ and $P(U > u) = 1 - u$ for any $u \in [0, 1]$. Thus,

$$F_Z(u) = P(Z \leq u) = P(1 - F_X(X) \leq u) = 1 - P(X \leq F_X^{-1}(u)) = 1 - F_X(F_X^{-1}(u)) = 1 - u. \tag{17}$$

Therefore, one can plot a histogram of the p-values of the incorrect annotations in order to visually verify that the p-values are well calibrated. The plots in Figure 15 show some visual verification of the p-values.

2.6 Comparison of score functions

Direct comparison of score function is not recommended because each scoring method along with their preprocessing steps and filtering are tuned and adjusted together. For instance, SPC-based score functions, such as the HyerpScore works well with TOP-N filter, but it likely would not work without this filter. Andromeda's score function also employs a TOP-N filter during its optimization procedure. In fact, notice that SPC-based score functions, such as those in Andromeda, HyperScore and OMSSA, do take into account peak intensities implicitly by using TOP-N filters. On the other hand, XCorr is optimized to work with unfiltered spectra and it is very likely that XCorr would not work with a TOP-N filter. XCorr uses a score calibration method, cross corre-

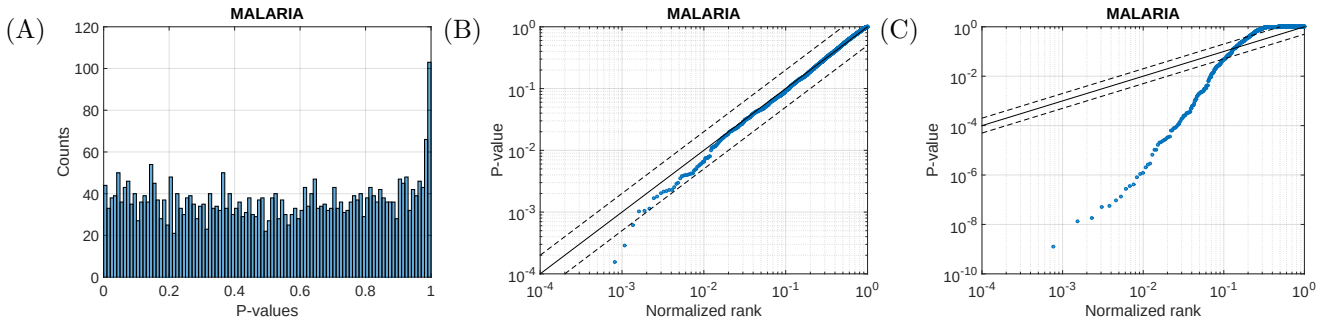

Figure 15: (A) Distributions of p-values of samples from a null distribution. The p-values were calculated with exact p-value method on the Malaria dataset. (B) The p-values from plot (A) are plotted against their normalized rank on a scatter plot. (C) The p-values produced with OMSSA on the Malaria dataset. This plot indicates that the p-value estimation by the OMSSA are liberal.

lation penalty, which gives an estimation on the null distribution. Therefore, TOP-N filter would remove noise and the null-distribution estimation would not be accurate anymore.

Some score function is defined with a combination of a score calibration. For instance, Andromeda and OMSSA directly estimates p-values, while the HyperScore does not include such step, therefore comparing HyperScore with the other score functions which include a score calibration would not be fair.

Several new score functions and database searching tools have also been introduced, including Mascot [7], X!Tandem[8], Morpheus [14], and MS Amanda [13]; however, these methods have resulted in only minor improvements as compared with SEQUEST [24].

2.7 Universality property

Different instruments, experimental protocols, and database-searching parameters have an impact on the experimental spectra observed. For instance, the ionization type could differ between instruments and, as a result, the experimental spectra might have different peak distributions. Furthermore, experimental protocols, the consideration of modifications or missed cleavages, influence the collection of theoretical peaks. Machine learning method trained on certain type of dataset might not necessarily generalize to other spectra generated with different types of instruments and experimental protocols. For instance, features learned from spectrum data obtained with high-energy collision dissociation (HCD) fragmentation may not be appropriate for data obtained with collision-induced dissociation (CID) or electron-transfer dissociation (ETD) fragmentation.

2.8 Learning new score functions

The peculiarity of learning score functions for spectrum annotation in this field is that it is not possible to obtain a human annotated spectrum dataset because human observers cannot go inside a mass spectrometer, visually observe the molecules and annotate the spectra they produce. Therefore, in this field, supervised training along with training-validation-test scenario is not applied; instead, machine learning (ML) methods are trained via self-supervision, in which a small data is annotated via standard database-searching methods and used as training data.

Since, the self-supervision does not involve human in the loop, this can be done for every data to be annotated. Therefore, the question is how well a method can generalize to obtain more annotations on the same dataset compared against the standard or other score functions. This approach was introduced in this field with Percolator in 2007 [32] and has become a standard since then; albeit it was introduced as a semi-supervised method.

Unfortunately, there is a potential danger with this approach in general. A ML method can learn to give preference to target peptides, that is a spectrum matched to target peptides can yield systematically higher scores than when it is matched to decoy peptides. This can result in a biased FDR estimation without showing any signs of problems. Therefore, it is important to show that the improvement in spectrum annotation made by a machine learning based scoring function does not arise from this bias.

Several methods have been proposed to replace the old-fashioned mathematical models in the data annotation pipelines shown on Figure 2 with data-driven machine-learning-based approaches at various stages and they indeed improved the annotation results significantly. For instance, we introduced two neural network based spectrum-peptide scoring methods [33] along with a new score calibration recently [5]. Percolator [32] is based on a linear SVM and it was introduced for re-scoring the spectrum annotations at stage (h), Prosit [34] is also a machine-learning-based post-processing method to give a more accurate spectrum annotation in stage (g), MS2PIP [35] is based on a machine-learning method to generate more natural-looking reference data for spectrum scoring at stage (b). However, the main limitation of these approaches is that these methods lie at different stages and they cannot access and combine information from other stages.

3 Evaluation of database searching results

After having run a database-searching method to annotate the experimental spectra with the top scoring peptide, the output results report some information about each annotation. The information includes an ID of the experimental spectrum, usually a scan ID, charge state, or file name, furthermore it includes the peptide sequence which produced the highest score among the candidate peptides. Furthermore, it includes p-values if available. The spectrum annotations are often ordered starting with the best scoring annotation going toward less likely correct annotations. Figure 16 illustrates a spectrum annotation result.

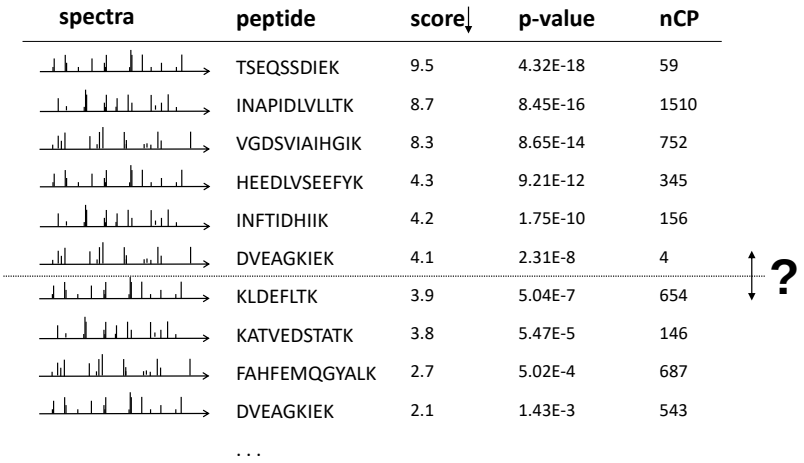

Figure 16: Illustration of the results of database-searching. Every experimental spectrum is annotated with the best scoring peptide from the corresponding set of candidate peptides. Each annotations is accompanied with the score, a p-value if available, and the number of the candidate peptides (nCP). The question is where to put the threshold to retain likely correct annotations?

In practice, it turned out that roughly 10-50% of the experimental spectrum can be annotated with high confidence, while the rest of the annotations are likely incorrect. Now, an interesting question is that how can we provide an accurate confidence about the annotations? Another questions is how to choose a threshold to decide which annotations to trust and keep and which annotations to discard? Moreover, how can we be sure that our error control procedure is correct?

The answer lies in the null-distribution again; however, this time we need the null-distribution of the scores of the incorrect spectrum annotations; that is, the top-scores of the incorrect peptide matches. We refer to this null distribution as experiment-specific null distribution (ESNL) to distinguish it from the spectrum-specific null-distribution (SSNL).

The score distribution of the spectrum annotations can be considered as a mixture model of two components, one standing for the scores of the correct and the other standing for the incorrect spectrum annotations. Figure 17A-B shows the XCorr and the p-value distribution of the spectrum annotations, respectively. In both plots, one can see two albeit overlapping modes for the scores of correct and incorrect spectrum annotations.

Our overall aim is to report a set of spectrum annotation which we deem as correct and trusted

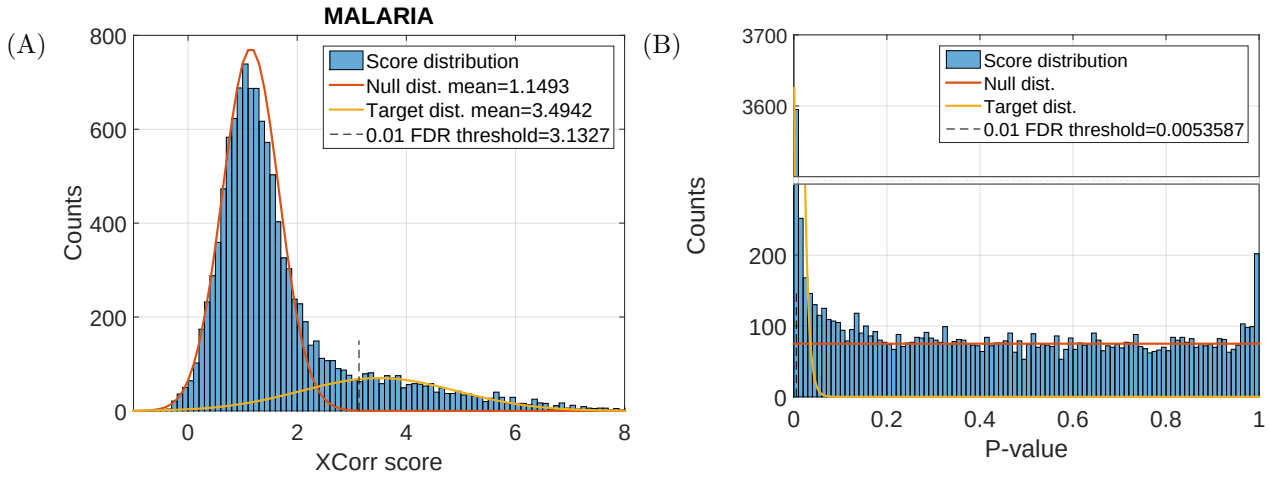

Figure 17: Distributions of XCorr scores (A) and the corresponding p-values (B). On the plot (A), we fitted a Gaussian mixture model of two components using the Expectation-Maximization method. On the plot (B), the models were fitted manually. Note that, these models on plots (A-B) are only for illustration, they do not represent the true models of the correct and incorrect annotations. Accepting spectrum annotations whose scores are above the thresholds t represented by a dashed line would provide an error of 1%. How did we get this t?

so that the error level in this set is around a specific, user-defined value. This is the false discovery rate (FDR), and we will discuss methods to control the FDR level in the next sections.

3.1 False discovery rate

The false discovery rate (FDR) with respect to a decision threshold t is, here, defined as the rate of the incorrect spectrum annotations, also called as false discoveries, among the total annotations which have matching scores greater than or equal than the threshold t. Here, we assume that a higher score indicates a better match. Formally,

$$FDR(t) = E\Big[\frac{D(t)}{T(t)} \mid T(t) > 0\Big] \cdot P(T(t) > 0), \tag{18}$$

where $D(t)$ (resp. $T(t)$) indicates the number of the incorrect (resp. correct) PSMs having a score larger than or equal to t. The other terms, $T(t) > 0$ and $P(T(t) > 0)$ are introduced to handle the $T(t) = 0$ situation.

In a typical mass spectrometry application, we usually want to select a threshold t so that the associated $FDR(t)$ of the spectrum annotation for a given experiment is at a certain level α. Visually speaking, the question is that where to place the vertical dashed threshold line in Figure 17 so that the rate of the incorrect spectrum annotations is at a level of α. Typical values for α are 0.001, 0.01, 0.05, or sometimes as high as 0.1. These levels often expressed in percentage such as 0.1%, 1%, 5%, 10%, respectively.

Let x indicate the scores of spectrum annotations (i.e. the scores of the top-scoring PSMs) and assume that larger score values indicate a better match. The FDR is known to have a Bayesian interpretation upon defining a two-component mixture model for the score density distributions

[36]:

$$f(x) = \pi_0 f_0(x) + (1 - \pi_0) f_1(x), \tag{19}$$

where f_0 is the density score distribution of the incorrect PSMs, $f_1(x)$ is the density score distribution of the correct PSMs, and π_0 is the fraction of incorrect PSMs. Such a mixture distribution can be seen in Figure 17A for the XCorr scores obtained with the Malaria dataset. This distribution has two modes, one taller at around XCorrs of 0.8-1.1 possibly corresponding to the distribution of scores of incorrect annotations f_0, and the second, wider mode at around XCorrs of 2.3-3.1 possibly corresponding to the distribution of the scores of the correct annotation f_1. The mixture distribution looks a bit different for p-values as it is shown in Figure 17B. The first mode at around 0.0-0.01 is corresponding to the p-value distribution of correct annotations f_1, while the p-value distribution corresponding to incorrect annotations f_0 is a uniform-like distribution over the interval $[0, 1]$.

Let $F(x)$, $F_0(x)$, and $F_1(x)$ are the complementary cumulative distributions, i.e. $F.(x) = \int_{X=x}^{+\infty} f.(X)dx$. Thus,

$$F(x) = \pi_0 F_0(x) + (1 - \pi_0) F_1(x), \tag{20}$$

and the FDR becomes

$$FDR(t) = \frac{\pi_0 F_0(t)}{F(t)}. \tag{21}$$

Now, let us discuss algorithmic methods to control the FDR at a certain, user specified level α.

3.2 FDR control with target-decoy approach

In order to control the FDR, we need to have an estimation on the number of the incorrect spectrum annotations. This can be estimated with using the so-called target-decoy-search strategy. This approach works as follows. Each peptide in the reference peptide dataset, that is associated to real, existing peptide molecule, is called the target peptide. For each target peptide, we generate another random peptide sequence called decoy peptide that does not exist in the reference peptide dataset. Therefore, the reference peptide dataset consists of two types of peptides: target and decoy peptides. The decoy peptides are often generated from target peptides via either (1) reversing the non-terminal amino acids, ot (2) shuffling the non-terminal amino acids. In order to obtain an unbiased FDR estimation with target - approach, the following criteria must meet:

1. We must ensure that the main characteristics of the set of target and decoy peptides are very similar, for instance, the amino acid frequencies, the distribution of the precursor ion mass, peptide length, should be the same among the candidate peptides.
2. We assume that for every spectrum annotation which is assigned to a decoy peptide there is another incorrect spectrum annotation assigned to a target peptide with roughly the same score.
3. The number of the target and decoy peptides must be equal, otherwise a correction factor should be employed in the FDR calculation.

4. It is assumed that incorrect spectrum annotations are equally likely to receive either target or decoy peptides.
5. The target and decoy peptides are distinct and independently generated.

Now, in the TDA the FDR is estimated by

$$\hat{FDR}(t) = \frac{D(t)}{T(t)}, \tag{22}$$

where $D(t)$ (resp. $T(t)$) indicates the number of the spectrum annotations which were assigned to decoy (resp. target) peptides having a score of t or higher. This FDR estimation can be rewritten:

$$\hat{FDR}(t) = \frac{\frac{D}{T}\frac{D(t)}{D}}{\frac{T(t)}{T}} = \frac{\hat{\pi}_0 \hat{F}_0(t)}{\hat{F}(t)}, \tag{23}$$

where D and T indicates total number of PSMs assigned to decoy and target peptides, respectively. Note that, the TDA approach includes implicitly the estimation of the proportion of incorrect PSMs among the targets. The Figure 18 separately shows the scores of spectrum annotations annotated with target or decoy peptides. Note that, the score distributions obtained with decoy peptides (brown) are used to model the experiment-specific null-distribution of the target peptides.

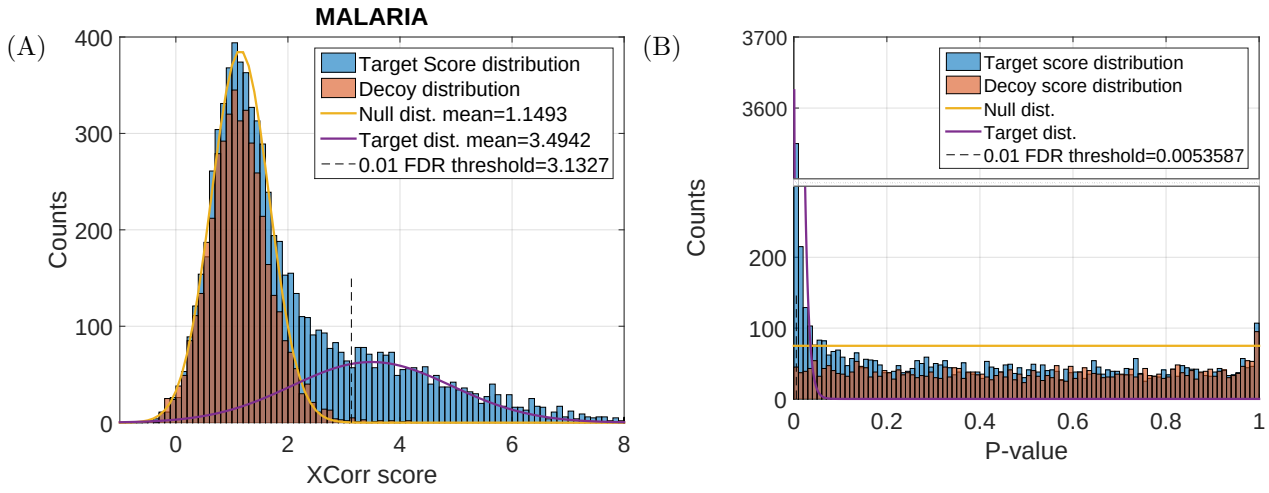

Figure 18: Distributions of target and decoy scores of XCorr (A) and the corresponding p-values (B). Note that the decoy score distribution is used to approximate the experiment-specific null-distribution for the spectra annotated with target peptides.

3.2.1 Q-values

The q-value of a spectrum annotation is defined as the smallest α level so that it is accepted at the α level of FDR. For instance, the q-value of a PSM is 0.005 then it is accepted at 0.5 % FDR level, but it is not accepted at, say, 0.50001 % FDR level. Note that the q-value of a PSM depends not only on the spectrum and its corresponding set of candidate peptides, but it also depends on other spectrum annotations too.

The pseudocode of q-values calculation algorithm is shown by Algorithm 2.

Algorithm 2: Q-value calculation using TDA.

Input : List of spectra along with their annotation and their accompanying scores : $\langle s_1, h_{1_j}, c_1 \rangle, \ldots, \langle s_n, h_{n_j}, c_n \rangle$

Output: $\langle s_1, h_{1_j}, c_1, q_1 \rangle, \ldots, \langle s_n, h_{n_j}, c_n, q_n \rangle$

```
Sort in decreasing order by the annotation scores c_i.
nTarget ← 0
nDecoy ← 0
for i = 1 → n do
    if h_{i_j} is target peptide then
        nTarget ← nTarget + 1
    else
        nDecoy ← nDecoy + 1
    end if
    q_i ← nDecoy / nTarget
end for
for i = (n − 1) → 1 do
    if q_{i+1} < q_i then
        q_i ← q_{i+1}
    end if
end for
```

The example of q-values calculation using Algorithm 2 is provided in Table 20.

spectra	peptide	score	p-value	nCP	FDR (D/T)	Q-value
	TSEQSSDIEK	9.5	4.32E-18	59	0/1=0	0
	INAPIDLVLLTK	8.7	8.45E-16	1510	0/2=0	0
	VGDSVIAIHGIK	8.3	8.65E-14	752	0/3=0	0
	HEEDLVSEEFYK	4.3	9.21E-12	345	0/4=0	0
	INFTIDHIIK	4.2	1.75E-10	156	0/5=0	0
	DVEAGKIEK*	4.1	2.31E-8	4	1/5=0.2	0.14
	KLDEFLTK	3.9	5.04E-7	654	1/6=0.16	0.14
	KATVEDSTATK	3.8	5.47E-5	146	1/7=0.14	0.14
	FAHFEMQGYALK*	2.7	5.02E-4	687	2/7=0.29	0.29
	DVEAGKIEK*	2.1	1.43E-3	543	3/7=0.42	0.42

Figure 19: Illustration of the Q-value calculation with TDA.

A typical results obtained with database-searching is often reported by the number of accepted spectrum annotations as a function of the q-values.

3.3 FDR control with P-values

When the spectrum annotations are accompanied with well-calibrated p-values (as shown in Figure 17B), the FDR can be controlled with the Benjamini-Hochberg (BH) procedure. This methods is based on that if all hypotheses were coming from the null distributions, i.e. all spectrum

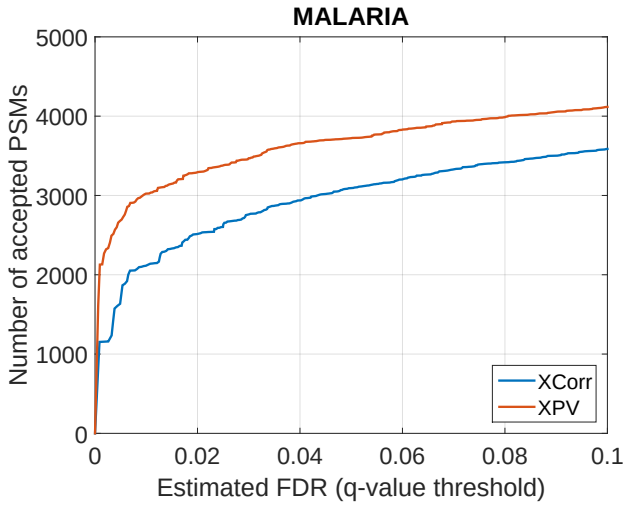

Figure 20: Illustration of the search results.

annotations were incorrect then the p-values would be uniformly distributed. Therefore,

$$p_1 \approx 1/m,\ p_2 \approx 2/m,\ \ldots,\ p_m \approx m/m, \tag{24}$$

where m denotes the number of spectrum annotations. Plotting the p-values against their ordered rank would yield a diagonal line as illustrated on Figure 21A. Let us choose a threshold t and suppose that there are T spectrum annotations whose accompanying p-values are less than t, that is $p_1, p_2, \ldots, p_T < t$. Note that we have $t \approx T/m$ from Eq. 24 and thus

$$p_i < t = T/m \qquad (i = 1, \ldots, T). \tag{25}$$

The number of incorrect spectrum annotations (w.r.t. t) is $D = T$, because we assumed all spectrum annotations are incorrect, thus $FDR(t) = D/T = 1$. Consequently, for a threshold $t = T/m$ we have 100 % FDR.

Let us suppose for a real experiment with m spectrum annotations in total, that there are C correct spectrum annotations whose accompanying p-values should be very small (something like 1e-43) and there are $m_0 = (m - C)$ incorrect spectrum annotations whose accompanying p-values are uniformly distributed. In the ranked list, the p-values of the correct annotations should

accumulate at the beginning followed by the p-values of the incorrect spectrum annotations:

$$p_{i_1} \approx 1/m_0,\ p_{i_2} \approx 2/m_0,\ \ldots,\ p_{i_{m_0}} \approx m_0/m_0, \tag{26}$$

where p_{i_j} denotes the p-values of the incorrect spectrum annotations. Putting back these p-values to the whole ranked list:

$$p_{i_1} \approx \frac{1}{m}\frac{m}{m_0},\ p_{i_2} \approx \frac{2}{m}\frac{m}{m_0},\ \ldots,\ p_{i_{m_0}} \approx \frac{m_0}{m}\frac{m}{m_0} \tag{27}$$

and for the first i_T p-values we have

$$p_{i_j} < t = \frac{T}{m}\frac{m}{m_0} \qquad (j = 1, \ldots, T). \tag{28}$$

This is illustrated in Figure 21B-D. We want to control the FDR at a certain level α; that is, we want $FDR(t) = \alpha$. Thus we need to adjust the threshold in Eq. 28 to

$$t = \frac{T}{m}\frac{m}{m_0} \cdot \alpha = \frac{T}{m}\frac{\alpha}{\pi_0}, \tag{29}$$

where we used the common notation $\pi_0 = m_0/m$.

The Benjamini-Hochberg (BH) protocol is based on the above and it goes formally as follows:

1. Order the spectrum assignments by their p-values in increasing order. Most significant spectrum assignments are at the beginning of the list.
2. Find the largest rank T so that the corresponding p-value $p_T < T/m \cdot \alpha/\pi_0$.
3. All spectrum assignment $p_1, p_2, \ldots, p_T$ are retrained and treated as trusted assignments at an FDR level of α, while all other assignments $p_{T+1}, \ldots$ are discarded.

If π_0 is not know and it is treated as $\pi_0 = 1$ then the BH protocol results in a conservative FDR estimation. In practical applications π_0 is unknown but it can be estimated [37].

3.4 Unbiasedness property of score functions

In order to obtain accurate FDR control and estimation, incorrect spectrum annotations ought to be assigned to either target or decoy peptides with equal likelihood. Standard raw scoring functions meet this condition because they do not have the capacity to distinguish between target and decoy peptides. However, machine-learning-based methods that involve target and decoy peptides in training to improve spectrum annotation accuracy can attain preference toward target peptides or annotations matched to target peptides. This results in a biased FDR estimation.

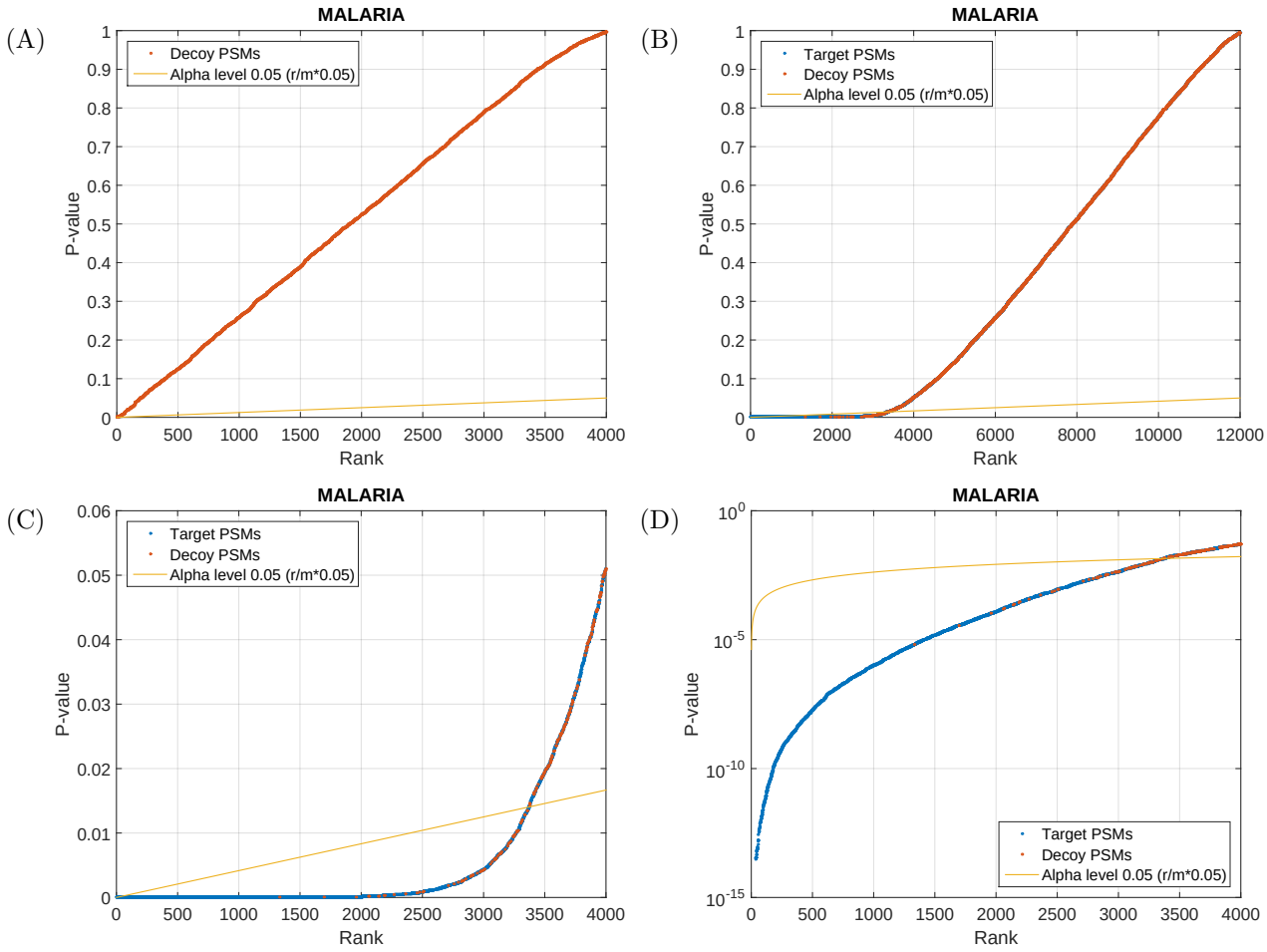

Figure 21: Illustration of controlling FDR with BH protocol. The plots shows the scatter plot of the p-values of spectrum annotations against their rank. P-values were obtained with exact p-value with the Malaria dataset. The blue dots correspond to target PSMs, red dots correspond to decoy PSMs. Note that BH does not require decoy annotations they are used solely for illustration. Yellow line indicates the threshold corresponding to the alpha level. Panel (A) shows the p-values of incorrect annotations plotted against their rank. Panel (B) shows the p-values of the correct and incorrect annotations plotted against their rank. Panel (C) is same as panel (B) but on a different range for visibility purposes. Panel (D) is the same as panel (C) but on a log scale.

spectra	peptide	score ↓	p-value	nCP	rank	r/m*α
	TSEQSSDIEK	9.5	4.32E-18	59	1	0.000909
	INAPIDLVLLTK	8.7	8.45E-16	1510	2	0.001818
	VGDSVIAIHGIK	8.3	8.65E-14	752	3	0.002727
	HEEDLVSEEFYK	4.3	9.21E-12	345	4	0.003636
	INFTIDHIIK	4.2	1.75E-10	156	5	0.004545
	DVEAGKIEK	4.1	2.31E-8	4	6	0.005455
	KLDEFLTK	3.9	5.04E-7	654	7	0.006364
	KATVEDSTATK	3.8	5.47E-4	146	8	0.007273
	FAHFEMQGYALK	2.7	5.02E-3	687	9	0.008182
	DVEAGKIEK	2.1	1.43E-2	543	10	0.009091
	IERQYTSAK	1.9	3.32E-2	154	11	0.01

Figure 22: Example for the BH control.

4 Error-tolerant search

Perhaps, one of the main drawback of the database-searching-based spectrum annotation is that spectra cannot be annotated if the corresponding peptide is missing from the reference dataset. In fact, roughly 20-40% of the spectra can be annotated with high confidence, all the other spectra is likely incorrectly annotated.

4.1 Missed and unexpected cleavages

The missed cleavages refers to the situation when the enzyme did not cleave the protein molecule at the predicted cleavage site. This might happen due to the fact that the cleavage site is inaccessible for the enzyme. This situation can be handled easy by simulating the missed cleavages, usually up to 2-3 missed cleavage sites in the in silico protein digestion. This step usually increases the number of the peptide sequences in the reference dataset by the 2-3 times.

The unexpected cleavage refers to the situation when the peptide molecule breaks into two-or-more parts, and the terminal of the results sub-peptides do not correspond to expected cleavage site. In order to handle this situation and include the corresponding peptide sequences into the reference data set, one needs to generate all peptides resulted from imperfect enzymatic digestion. The number of the peptides in the reference dataset for various in silico peptide generation is shown in Table 5

Original protein fragment
MEICRGLR|**SHLITLLLFLFHSETICRPSGR**|**K**|**SSK**|**MQAFR**|**IWDVNQK**|**GACEEFQWK**...
Tryptic peptides
MEICRGLR,**SHLITLLLFLFHSETICRPSGR**, **K**, **SSK**, ...
Missed cleavages = 1
MEICRGLRSHLITLLLFLFHSETICRPSGR, **SHLITLLLFLFHSETICRPSGRK**, **KSSK**, **SSKMQAFR**, ...
Non-enzymatic peptides (spontaneous breakdown)
EICRGL, **EICRG**, **EICR**, ... **WDVNQ**, **WDVN**, **DVNQ**, **VNQ**, ...
Modified peptides
M(ox)EICRGLR, **M(ox)QAFR** ...

Table 4: Illustration of protein sequence digestion in silico. The protein is taken from International Protein Index *IPI:IPI00000045.1—SWISS-PROT:P18510-1*. Vertical bars mark tryptic cleavage sites. **(ox)** indicates the oxidation on methionine (M).

4.2 Modifications

Modifications can occur to peptide molecule. The typical modifications are (a) post-translational modifications (PTMs) which regulate the protein activity and function in vivo by attaching a small molecule, such as phosphor to certain amino acids or (b) chemical modifications indicates a modification when a small atom or molecule is attached to certain amino acids, for instance, an oxygen atom can attached to the methionine amino acid during sample preparation, and finally (c) modifications may include amino acid mutations as well. For computational data analysis,

Table 5: The number of the unique peptides in the reference peptide dataset

	Tryptic	Semi-tryptic	Non-Tryptic
Target	221,900	3,742,175	97,404,383
Decoy	220,360	3,693,799	96,681,819
Total	442,260	7,435,974	194,086,202

Table 6: The numbers of a modified peptides by the number of allowed variable modifications.

No. PTMs included (K)	0	1	2	3	4	5	6	7	8	9
No. modified peptides	1	55	275	20,625	206,250	1,443,750	7,218,750	25,781,250	64,453,125	107,421,875

The number of modified peptides were calculated for one peptide with 11 amino acids, and assuming that each amino acid can be modified by 5 different PTMs. The general formula to calculate the is $\binom{L}{K}M^K$, where L, K, and M denote the length of the peptide, the number of variable modifications to be included to the peptide, and the average modifications per amino acids.

these modifications can be handled and identified with the same algorithms, therefore we refer to them as modifications and it is commonly abbreviated as PTM. To generate modified peptide sequences is straightforward; however, the number of modified peptides can grow combinatorially. The Table 6 shows the number of the modified peptide sequences for various number of PTMs allowed to be present in the modified peptides at the same time. The peptide sequence is of length 11 and we assumed that all amino acid can be modified by 5 different PTMs. In the table one can see that there are 55 modified peptides which contain 2 PTMs and there are 275 modified peptides with 3 PTMs. The number of modified peptides also grows combinatorially with respect to the allowed PTMs.

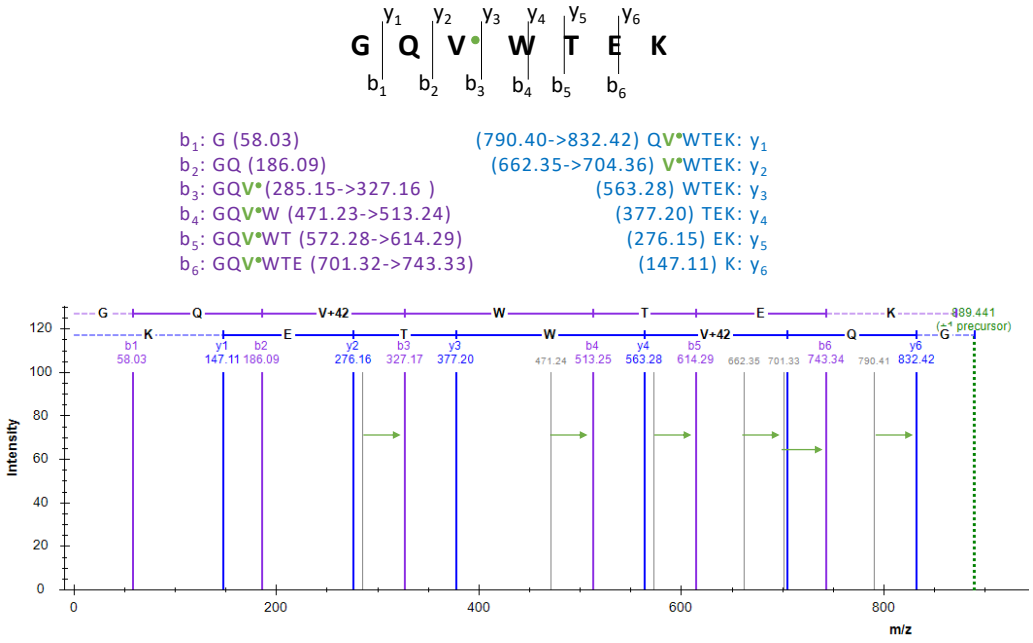

Figure 23: Illustration of the changes induced by modifications. A modification changes the mass of the standard amino acids and it results in shifting the peaks of the primary fragmentation ions which contain the modified residue. The

The main computational approaches for PTM identification are the following.

- **Targeted PTM identification.** In this approach, the experimenter has to guess few

modifications which might be in the sample and specify this modifications individually. All major search engines, e.g. Crux, Sequest, Mascow, X!Tandem, Andromeda, supports this

- **Untargeted PTM identification.** This approach employs a large collection of know modifications (PTM DB) and it uses some heuristics method to find PTMs by avoiding the combinatorial explosion of by generating all possible modified peptide sequences.

- **De novo PTM identification**. This approach does not use any know modifications as reference, but it tries to identify modification in the sample by searching for certain regular patters in the spectrum data.

5 Large search space

There are two major challanges when one generates too many peptides with either (1) too many modifications, (2) using non-standard digestion rules such as semi- or non-tryptic digestion, or (3) combining fasta files of too many taxons.

One of the main challenges is simply computational, i.e. the spectrum annotation procedure might need significantly more time to match every spectrum against thousands of millions of candidate peptides. As discussed in the previous section the number of semi- or non-tryptic peptides as well as the number of the modified peptides can explode combinatorially.

The other main challenge is that spectrum annotation undergoes a sort of multiple testing correction; thus, a high score may not end up being significant and in turn it results in fewer number of annotations at any level of FDR. We discuss this in details for similarity like scores (such as XCorr) and for p-values.

For the first case, let us consider the scoring with a similarity-like score function such as XCorr. Furthermore, consider an experimental spectrum s is matched against its candidate peptides $CP(s)$ but none of the peptides are related to s. The general problem arises from the fact that matching an experimental spectrum s against unrelated peptides from CP can be treated as random sampling. Thus, the more unrelated peptides are in CP, the higher the top-score (i.e. maximum score) of these PSMs, in some cases, a top-scoring unrelated peptide may produce even a higher score than the related, correct peptide. Now, consider a situation when a set of m experimental spectra is independently searched against two peptide reference data sets: first, it is searched against a relatively small set with mostly relevant peptides D_1 and, second, it is searched against an inflated peptide dataset D_2 with many unrelated peptides. Formally, we assume that $CP_1(s) \subset CP_2(s)$ and $CP_2(s)$ is significantly bigger than $CP_1(s)$ for any spectra s, where $CP_i(s)$ denotes the set of candidate peptides selected from D_i. For instance, D_1 can be a set of fully tryptic peptides, D_2 can be the set of non-tryptic peptides. The two score distributions of the correct annotations from the two searches remain roughly the same (similar mean, modes, and tails), because the number of the matching peaks of the correct spectrum annotations does not depend on the size of CP_i or the D_i; however, one can expect more correct spectrum annotations from the search against D_2; because it might contain the correct (non-tryptic) peptides for some

experimental spectra, which are not included to D_1.

More interestingly, the two score distributions of the incorrect spectrum annotations might differ considerably, because the scores of the incorrect annotations from the second search (against D_2) are greater than or equal to the scores obtained from the first search (against D_1). Roughly speaking, the null distribution from the second search is a bit right to the null distribution of the first search, and searching against a large reference peptide set "shifts" null distribution "rightwards" closer to the score distribution of the correct annotation. As a consequence, one needs to use a higher threshold in order to control the FDR at the same level. Now, the question is that whether one can gain more correct spectrum annotation from the second search than one looses due to the increased acceptance threshold.

This is illustrated in Figure 24. The malaria dataset was searched against a series of seven, increasingly growing CP sets and the score histograms of the target and decoy annotations were plotted along with the acceptance threshold for 1 % FDR. The specification of constructing CP sets is given in the titles of the plots and description can be found in the footnote of the figure. By observing the series of the score histogram through panels (A-G) one can see that how the mode of the null distribution is "shifted" from around 1.3543 to 2.2797 and the decision threshold to maintain 1 % FDR level also increased from 3.1281 to 4.7184. This resulted in a significant drop in the number of accepted spectrum annotations from around 2000 to around 1000 at 1 % FDR. The number of accepted spectrum annotations as a function of estimated FDR levels is also shown in panel (H).

For the second case, let us now consider the scoring evaluated with well calibrated p-values. The experimental spectra are annotated with the peptide producing the most significant p-values, i.e. the smallest p-value, and they must undergo a Sidak-correction in order to ensure that the experiment-specific null distribution remains uniformly distributed. However, when an experimental spectrum is annotated with the correct peptide, its accompanying p-value also undergoes the Sidak correction. For instance, if a spectrum annotated with the best scoring peptide which produces a p-value of 1e-13 out of, say, 1325 candidate peptides then the Sidak corrected p-value becomes 1.3254e-10; i.e. the p-value is increased by a three order of magnitude and it might not pass the decision threshold anymore.

This is illustrated in Figure 25. The experiments are the same as above in Figure 24, but now the p-values are used to evaluate the search results and control the FDR. The panel (A) shows that how the p-values are inflated as the number or the candidate peptides grow. The panel (B) shows the number of accepted spectrum annotations as a function of the q-values. In both panels one can see that the number of accepted spectrum annotations at 1 % FDR level drops from nearly 3000 to around 1000.

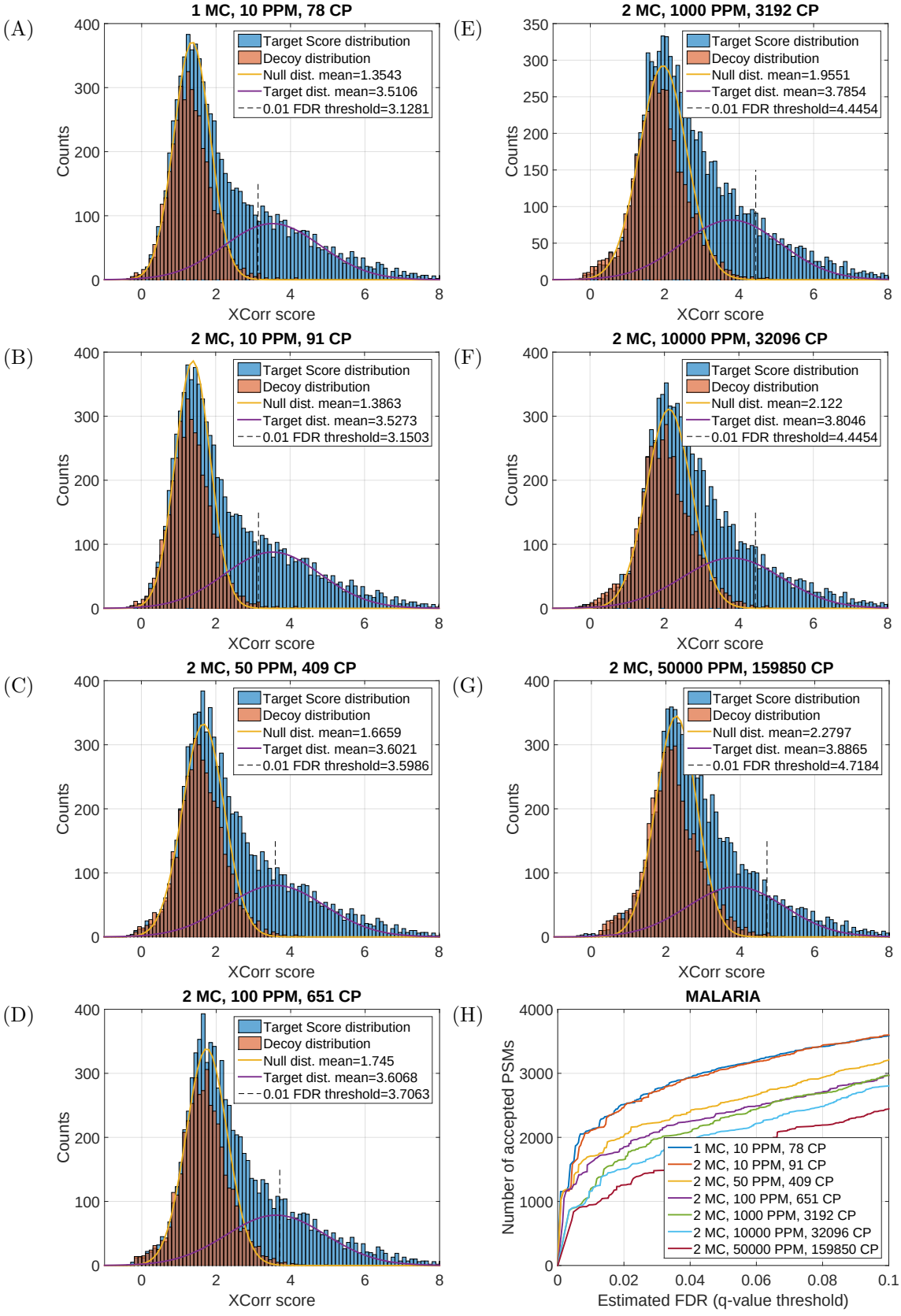

Figure 24: Illustration of the effects of the size of the CP on the score histograms and on the decision thresholds. The main parameters to generate CP are given in the plot titles. MC means the number of the missed cleavages, PPM indicates the precursor ion mass tolerance window in ppm, and CP shows the average size of the CP sets. The Gaussian models of the null and target distributions (solid lines) were approximated with EM and shown only for illustration; (they are not the true models). The Panel (H) shows the number of spectrum annotations at various FDR levels obtained from data from panels (A-G).

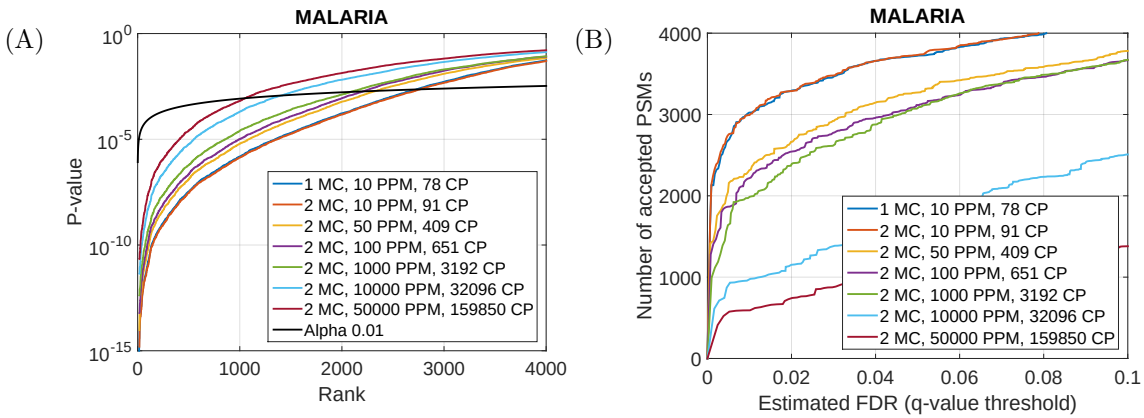

Figure 25: The effects of the size of the CP set on the search results using p-values with the Malaria dataset. (A) Scatter plot of the p-values of spectrum annotations against their rank obtained in different search settings. MC means the number of the missed cleavages, PPM indicates the precursor ion mass tolerance window in ppm, and CP shows the average size of the CP sets. The Panel (B) shows the number of spectrum annotations at various FDR levels obtained from data from various search settings for comparisons.

6 Main developed methods

6.1 Learning of new score functions

6.1.1 The BoltzMatch method

We introduced a novel method to learn score functions utilizing restricted Boltzmann machines (RBMs) in order to enhance the discriminative power of the score functions. RBMs are stochastic, fully connected neural networks [38] that pioneered the deep learning by being proposed as the building blocks of deep belief networks and that achieved state-of-the-art performance in various fields. In our approach, called BoltzMatch, we model the joint probability of observing an experimental s_i and a theoretical h_j spectra via RBMs, defined as

$$p(s_i, h_j) = \frac{1}{Z} \exp\{E(s_i, h_j)\}, \tag{30}$$

where the theoretical spectrum h_j is treated as an unobservable latent variable, an idealized version of the observed, flawed experimental spectrum s_i, which contains unexplainable peaks and incomplete fragmentation ion series. $E(s_i, h_j) = s_i^T W h_j$ is referred to as an energy function, and Z is a normalization factor[2], in which the parameters in W are to be learned from the observed mass spectrometry data. The log-likelihood $\log p(s, h) = E(s, z) - \log Z$ remarkably resembles the XCorr function defined in Eq. 5. On the one hand, one can roughly regard the XCorr as a log-likelihood of a manually crafted RBM, while on the other hand, one can roughly regard BoltzMatch as a generalization of XCorr in which the parameters are learned from the data. A graphical illustration of XCorr and BoltzMatch are shown in Figure 27.

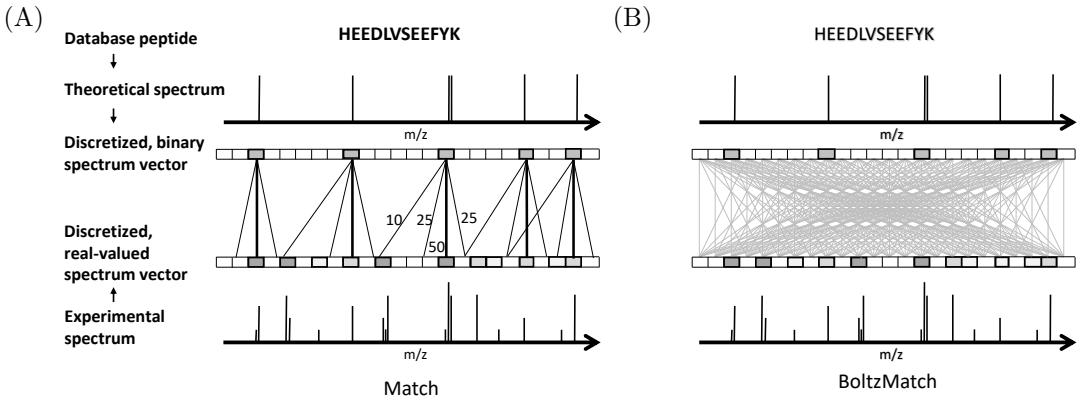

Figure 26: Graphical models of XCorr and BoltzMatch score functions along with their parameterization. (A) XCorr weights matching ions by 50, flanking peaks by 25, and losses by 10; the weight values were specified manually. (B) Fully connected stochastic neural network, BoltzMatch, for matching observed spectra with theoretical ones. BoltzMatch considers the association between all peak pairs and learns to weight them solely from the data.

By our experiments, we showed that BoltzMatch learns chemically explainable patterns among peak pairs of the observed and theoretical spectra, and as an outcome it may augment peaks

[2]defined as $Z = \sum_{s',h'} \exp\{E(s', h')\}$ for all possible vectors s', h'.

depending on their semantic context or even reconstruct peaks of unobserved but expected fragmentation ions during its internal scoring mechanism. This information is incorporated into the scoring, which results in an increased power in discriminating between correct and incorrect spectrum annotations. Additionally, BoltzMatch does not require manual instrument-specific and experiment protocol-based parametrization such as the specification of the secondary fragmentation ions such as a ions, nor does it need manual weight calibration for the matching peaks (unlike XCorr). As a result, BoltzMatch annotates 30-50% more spectra than XCorr.

The results were published in the article entitled

> **Identification of tandem mass spectrometry data using stochastic neural networks** written by Pavel Sulimov, Anastasia Voronkova, Attila Kertész-Farkas and which appeared in **Bioinformatics** 2020, 18(5), 2354–2358

and it can be found in the Appendix of the main dissertation text.

6.1.2 The Diversifying Regularization method

The diversifying regularization (DR) was introduced as a general regularization method to help train arbitrary deep generative and discriminative models.

We define the regularization term by divergence functions over the distribution functions over the latent variables. For instance, for two given data $\mathbf{x}_p$ and $\mathbf{x}_q$ the regularization is defined as $D\left(Q^{\phi}_{\mathbf{x}_p},\ Q^{\phi}_{\mathbf{x}_q}\right)$. This term can be included to the MLE as follows:

$$r(\theta, \phi; \mathcal{D}) = \sum_{\mathbf{x} \in \mathcal{D}} \log \sum_{\mathbf{h}} P(\mathbf{x}, \mathbf{h}; \theta) + \alpha \sum_{(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{N}} D\left(Q^{\phi}_{\mathbf{x}_p},\ Q^{\phi}_{\mathbf{x}_q}\right), \tag{31}$$

where r stands for regularized likelihood. The first term is the log-likelihood to be maximized. The second term is the diversifying regularization which introduces a penalty on two distributions if they are similar, but they should not be, and α is a trade-off parameter. D denotes a divergence function for probability distributions, for instance, D can be defined as follows:

$$D_H(Q^{\phi}_{\mathbf{s}_p}, Q^{\phi}_{\mathbf{s}_q}) = 1 - \sum_{h=\{0,1\}} \sqrt{Q^{j,\phi}_{\mathbf{s}_p}(h) Q^{j,\phi}_{\mathbf{s}_q}(h)} \tag{32}$$

A good source of divergence functions is provided by, e.g., Csiszár's f-divergence class [39].

The DR method turned to be essential tool to train the BoltzMatch method for spectrum annotations. Experimental spectra can contain ubiquitous peak which appear in almost every spectrum at the same m/z location. For instance, when samples were prepared using TMT6-plex labeling which has an associated weight of 229.16293 Da, then one can observe peaks around 230 m/z and 115 m/z in almost all experimental spectra. These peaks possibly correspond to single charged and double charged TMT labeling residues. These ubiquitous peaks do not contain useful information for spectrum identification but they interfere in generative modeling as they can correlate with all other peaks. To mitigate the effect of these ubiquitous peaks, we employed

the DR method in the following form:

$$DR = \sum_{s_i, s_j \in MB} h_i^T h_j, \tag{33}$$

to the learning objective defined in Eq.30, where s_i, s_j are observed spectrum pairs from a given mini-batch MB and $h_i \sim p(h_i \mid s_i) = \sigma(s^T W)$ (h_j is defined similarly), where $\sigma(a) = (1 + \exp(-a))^{-1}$ is the sigmoid function.

The article about the diversifying regularization (DR) is entitled

> **Guided Layer-wise Learning for Deep Models using Side Information** written by Pavel Sulimov, Elena Sukmanova, Roman Chereshnev, Attila Kertész-Farkas* and which appeared in **Communications in Computer and Information Science**, 2019, 30(2): 234-241

and it can be found in the Appendix of the main dissertation text.

6.1.3 The Slider method

The training of BoltzMatch is cumbersome even with the application of DR because the BoltzMatch model is very large. It contains 4 million parameters for low-resolution fragmentation settings (LRFS) and 1.6 billion parameters for high-resolution fragmentation settings (HRFL), so training becomes time-consuming and the model becomes too prone to overfitting. For HRFS, the BoltzMatch model does not fit in the memory of common GPUs that are commercially available nowadays so the training must be done with standard CPUs.

Therefore, we constructed a deep convolutional neural network architecture (ConvNet) [38] for scoring peptide-spectrum matches PSMs. The main components of ConvNets are kernels that are essentially sliding windows with trainable parameters; therefore, we refer to the ConvNet architecture as Slider. Slider can be thought of as a generalized version of the XCorr scoring method with more than one sliding window in which the parameters are learned from the observed spectrum data. A graphical illustration of Slider is shown in Figure 27B.

The training of Slider is stable and it can be trained with relatively small datasets because of the following three reasons. First, the kernels of Sliders are narrow, meaning they cover only the range of ±10.0 Dalton in the experimental spectra and we suspect that the patterns to be learned are simple. Second, the total number of trainable parameters of Slider are 465 for the LRFS and 4105 for the HRFS, respectively, which can be considered relatively small compared to deep learning architectures (having millions of parameters) employed in computer vision. Third, as Slider being a fully ConvNet, it shares weights at each spectrum bin position. This means that Slider can be considered as a fully connected neural network which aims to predict the presence of a theoretical fragmentation ion at any positions from a relatively small range of the experimental spectrum. That is, Slider uses the same parameters to predict an ion at position, say, 467 from a range [456,477], and at a position, say, 1723 from a range of [1712,1733] for LRFS. This further implies that, if there are 2000 bins in an experimental spectrum then it would provide 2,000

training instances for LRFS. Thus, if the training dataset consists of 10,000 MS/MS experimental spectra, then there are 2 million training instances per 465 trainable parameters for LRFS, and there are 40 million training instances per 4105 parameters for HRFS.

Our conclusions about the experimental results are that Slider has a slightly less discriminative power than BoltzMatch under strict FDR control (around 0.1%) with low- or high-resolution fragmentation settings. This, however, can be attributed to the fact that BoltzMatch contains 17 thousand times more trainable parameters than Slider (BoltzMatch: 4 million; Slider: 465) for LRFS, and 379 thousand times more (BoltzMatch: 16 Billion; Slider: 4,105) for HRFS. Nevertheless, Slider slightly outperforms the BoltzMatch at FDRs higher than 0.5%. We speculate that the convolutional kernels learn isotopic patterns since the they weight adjacent elements in a relatively narrow, ± 10.0 Da sliding window.

Perhaps the most interesting observation is that Slider with LRFS can provide almost as many spectrum annotations as Slider or other methods with HRFS. Investigating our experimental results shows that Slider with LRFS provides only 4.1%, 2.5%, and 2.3% fewer PSMs at 1% FDR, but is 8.3, 12.9, and 1.08 times faster than Slider, BoltzMatch, and XCorr with HRFS, respectively.

Slider, similarly to BoltzMatch learns an optimal feature extraction for the spectrum data without human intervention in order to achieve the best performance in spectrum annotation without requiring manual instrument-specific or experiment protocol-based parametrization.

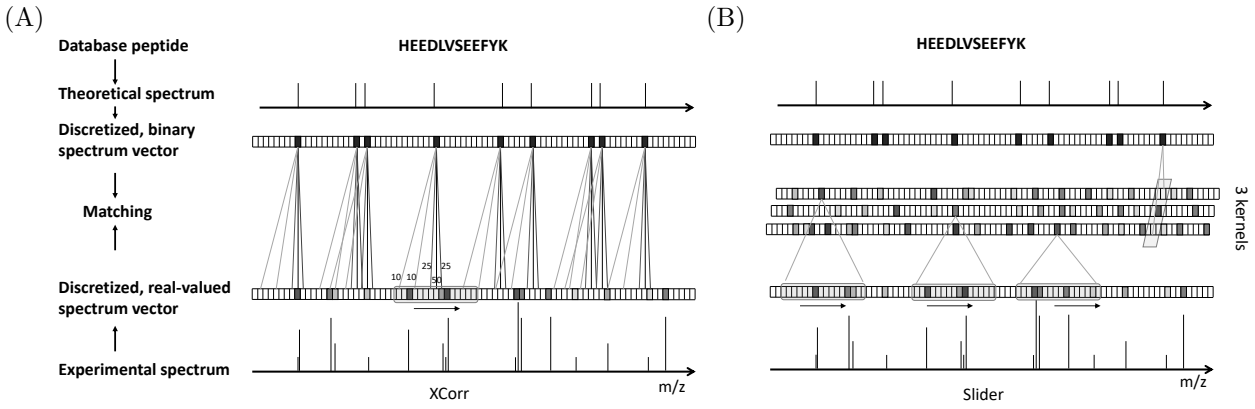

Figure 27: Graphical models of XCorr and Slider score functions. (A) XCorr weights matching ions by 50, flanking peaks by 25, and losses by 10; the weight values were specified manually. (B) Slider is a deep convolutional neural network in which the kernels act as weighted sliding windows and the weights are optimized from the MS/MS data in an end-to-end learning fashion.

The results were published in the article entitled

> **Deep convolutional neural networks help scoring tandem mass spectrometry data in database-searching approaches** written by Polina Kudriavtseva, Matvey Kashkinov, Attila Kertész-Farkas and which appeared in **Journal of Proteome Research** to appear in 2021,

and it can be found in the Appendix of the main dissertation text.

6.2 Statistical methods in improve the power of annotation methods

6.2.1 The Tailor method

The peptide-spectrum-match (PSM) scores obtained with machine learning-based methods, as described above, turned to be spectrum-dependent meaning that a matching score of t might indicate a significant match to one spectrum and insignificant one to another spectrum. We introduced a rapid and a non-parametric method, called Tailor, for fast score calibration which does not involve optimization.

Essentially, PSM scores are calibrated to the spectrum-specific null distributions, i.e. these methods aim at providing some statistical p-value or a $-logE$ value estimation as discussed in chapter 2.4. The success of the score calibration methods relies on how well they approximate the tail or the extreme tail of the null distribution to obtain a p-value estimation. Some methods are mode-seeking, meaning they fit the modes of the model and the empirical null distribution; however, they might result in an inaccurate fit at the tails. Other methods aim at modeling the tail exactly but they do so at the cost of computation time.

We developed a new, heuristic score calibration method, called Tailor, which calibrates the score of a PSM to the last 100-quantile, Q100, at the tail of the empirical null distribution, which is constructed for a given experimental spectrum from the scores obtained during scoring it against the candidate peptides. The Tailor method exploits the tail of the observed null distribution, where random scores are observed during the database search step, but not the extreme tail, where samples are rare. This is in contrast to the exact p-value (XPV) methods (MS-GF+), which enumerate all random scores, including those at the extreme tail, at the expense of the CPU time to obtain an exact and accurate empirical null distribution. Therefore, Tailor is quick and works with any score function, albeit less accurate (i.e., a heuristic approach), whereas exact methods are accurate, albeit slow and require specific score functions.

Experimental results proved Tailor to be a powerful score calibration method. It provides nearly as many spectrum annotations as the computationally exhaustive methods such as exact p-value; albeit at a fraction of the time.

The results were published in the article entitled

> **Tailor: non-parametric and rapid score calibration method for database search-based peptide** written by Pavel Sulimov, Attila Kertész-Farkas and which appeared in **Journal of Proteome Research** in 2020, 18(5), 2354–2358

and it can be found in the Appendix of the main dissertation text.

6.2.2 Bias evaluation in spectrum annotation

Accurate target-decoy-based FDR control of spectrum annotation relies on an important but often neglected assumption that incorrect spectrum annotations are equally likely to receive either target or decoy peptides. We showed that this assumption is often violated in practice by popular methods. Preference can be given to target peptides by biased scoring functions, which result in

liberal FDR estimations, or to decoy peptides by correlated spectra, which result in conservative estimations.

In particular, we showed that Percolator is able to discriminate certain types of decoy peptides from target ones by exploiting information from certain leaky features which can leak information about the types of the peptides. This is demonstrated in Figure 28.

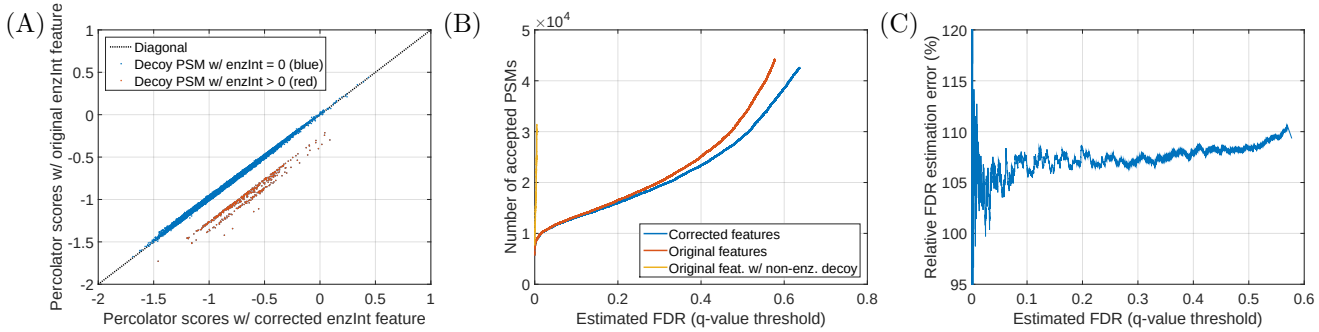

Figure 28: Bias in Percolator's features. (A) Percolator scores of decoy PSMs obtained with original features vs corrected enzInt features. (B) Percolator's results using default settings (red), using corrected enzInt features (blue), and using non-enzymatic decoy peptide database (yellow). (C) Estimated error, induced by enzInt feature, in FDR estimation at various levels, calculated based on the horizontal difference between the red and blue curves from panel (B).

Furthermore, scoring functions can give preference to target (or decoy) peptides during the spectrum identification search without even considering peptide labels. Contrary to expectations, the distribution of the theoretical target and decoy spectra (i.e. spectra generated from peptide sequences in silico) is slightly different in the spectrum vector space, and a simple linear model can exploit this information. For instance, a logistic regression (LogReg) achieved a 0.551 AUC score on classifying the theoretical target and decoy peptides. The result means that scoring functions that take into account peak location specific weights can induce bias, whether the weights are tuned manually or are learned by a particular machine learning algorithm. The situation with vanilla artificial neural networks (ANNs) is even more dismal (or astonishing). In the same dataset, an ANN achieved a spectacular AUC score as high as 0.902 in peptide classification. See blue and red ROC lines in Figure 29A. This means that scoring functions that account for peak pair and peak location-specific weights can induce large biases. Perhaps deep learning methods may achieve better discrimination between target and decoy peptides. However, when the target peptides are split randomly into positive and negative sets, the ANN achieves an AUC score of only 0.543. In our opinion, this shows that the distributions of the target and decoy spectra are indeed different, and ANN does not achieve a high AUC score due to data memorization. In practice, for instance, the DRIP scoring function [40] can give preference to target peptides.

Lastly, we discussed the possible consequences of correlated theoretical target and decoy peptides. For instance, the target FYDDENLTE and its reversed FTLNEDDYE peptides generate exactly the same theoretical spectra and the same matching scores against the peptide's experimental spectrum, resulting in a high-scoring decoy PSM. Consequently, we argued that statistics involving high-scoring correlated decoy peptides either implicitly, as in the ratio of the top two scores (Δc_n), or explicitly, as in separated target-decoy search, can result in a conservative FDR

estimation yielding fewer spectrum annotations at various FDR levels (q-value thresholds). Furthermore, the effect of the correlated target-decoy peptides is more enriched in small proteome datasets with high precursor information, because (a) correlated decoy peptides are always present among candidate peptides and (b) they face less competition from few others due to data sparsity.

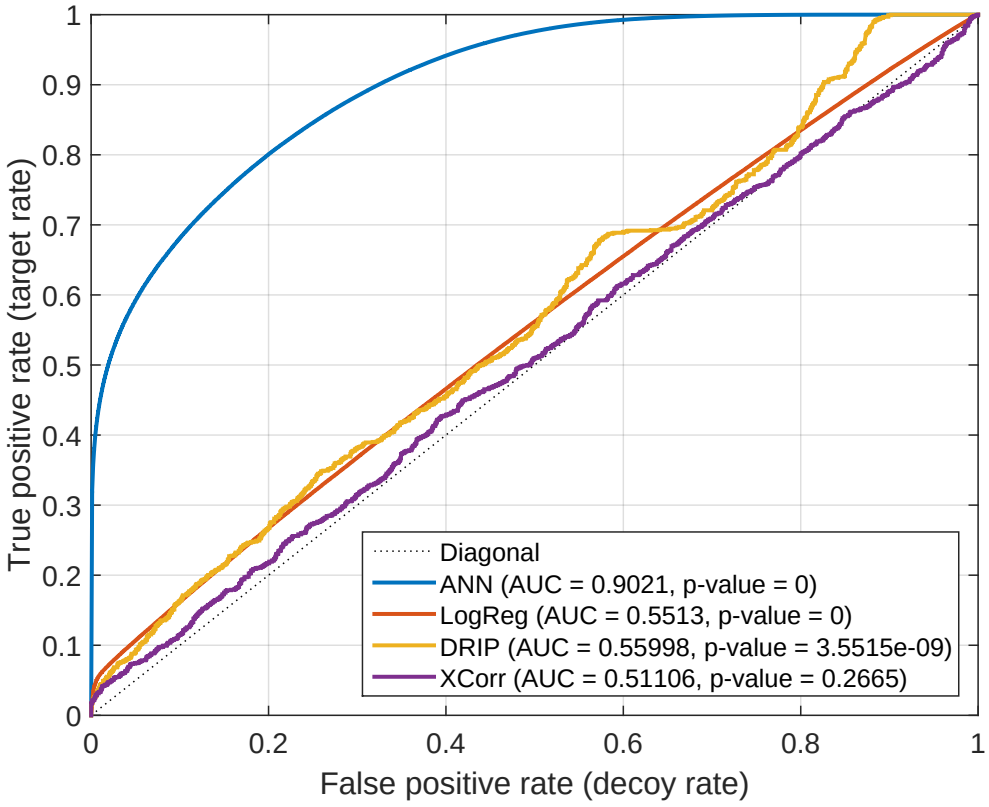

Figure 29: Discrimination of target and decoy peptides. (A) Discrimination of target against decoy peptides with ANN (blue), logistic regression (LogReg, red), DRIP (orange), and XCorr scoring function (purple) evaluated by ROC analysis. The diagonal line (dashed line) indicates an unbiased scoring function and identical distributions of the target and decoy PSM scores. P-values of ROC analyses were obtained with two-sided Mann-Whitney U-test.

The results were published in the article entitled

> **Bias in false discovery rate estimation in mass-spectrometry-based peptide identification** written by Yulia Danilova, Anastasia Voronkova, Pavel Sulimov, Attila Kertész-Farkas and which appeared in **Journal of Proteome Research**, 2019, 18(5), 2354–2358

and it can be found in the Appendix of the main dissertation text.

6.2.3 The Cascaded search method

Accurate assignment of peptide sequences to observed fragmentation spectra is hindered by the large number of hypotheses that must be considered for each observed spectrum. A high score

assigned to a particular peptide-spectrum match (PSM) may not end up being statistically significant after multiple testing correction. Researchers can mitigate this problem by controlling the hypothesis space in various ways: considering only peptides resulting from enzymatic cleavages, ignoring possible post-translational modifications or single nucleotide variants, etc. However, these strategies sacrifice identifications of spectra generated by rarer types of peptides.

We introduced a statistical testing framework, cascade search, that directly addresses this problem. The resulting cascade search algorithm operates on an ordered series of peptide groups, similar to ISPTM [41] and stratified search. Cascaded search requires that the user specify a priori a statistical confidence threshold as well as a series of peptide databases. For instance, such a cascade of databases could include fully tryptic, semitryptic, and nonenzymatic peptides or peptides with increasing numbers of modifications. Cascaded search then gradually expands the list of candidate peptides from more likely peptides toward rare peptides, sequestering at each stage any spectrum that is identified with a specified statistical confidence. However, whereas ISPTM treats each spectrum independently, thereby failing to control the FDR in the reported list of optimal PSMs, cascade search takes into account the entire collection of spectra to exert multispectrum FDR control.

In our experiments, we compare cascade search to a standard procedure that lumps all of the peptides into a single database, as well as to a previously described group FDR procedure that computes the FDR separately within each database. We demonstrated, using simulated and real data, that cascade search identifies more spectra at a fixed FDR threshold than with either the ungrouped or grouped approach. Cascade search thus provides a general method for maximizing the number of identified spectra in a statistically rigorous fashion. Cascaded search is presented in Algorithm 3.

Algorithm 3: Controlling FDR with cascaded groups. The input is a collection S_0 of spectra, a series $D^1, \ldots, D^n$ of peptide databases, an FDR threshold α, and a threshold k specifying the minimum number of identifications required per group.

Input : $S_0, D^1, \ldots, D^n, \alpha, k$
Output: set of PSMs R with the annotated spectra at FDR rate of α
  $R \leftarrow \oslash$
  for $i \leftarrow 1 \ldots n$ **do**
    $(M^i, C^i, E^i) \leftarrow \textsc{Search}(S^{i-1}, D^i)$
    $P^i \leftarrow \textsc{CalculatePValues}(S^i, M^i, C^i)$
    $A^i \leftarrow \textsc{ControlFDRbyBH}(P^i, \alpha)$
    if $|\{i \mid a_j^i = 1\}| < k$ **then**
      break
    end if
    $R \leftarrow R \cup \{(s_j^{i-1}, e_j^i, p_j^i) | a_j^i = 1\}$
    $S^i \leftarrow \{s_j^{i-1} | a_j^i = 0\}$
  end for
  Return R

The results were published in the article entitled

Tandem Mass Spectrum Identification via Cascaded Search written by Attila

Kertész-Farkas, Uri Keich, William Stafford Noble and which appeared in **Journal of Proteome Research**, 2015, 14(8), 3027–3038

6.2.4 The Mix-Max method

At least four distinct, decoy-based FDR estimation protocols have been advanced in the literature. The first, proposed by Elias and Gygi [42], finds the best matching peptide for each spectrum relative to a concatenated target-decoy database and estimates the FDR among all peptide-spectrum matches (PSMs) above a specified score threshold. In some cases, the top score is less than a specified score threshold, in which case no peptide is indicated. A key component of the Elias and Gygi strategy is *target-decoy competition* (TDC), in which the top-scoring target and decoy peptides compete with one another and only the higher-scoring of the two peptides is retained in the final list. In practice, this competition is carried out by searching the spectra against a concatenated database containing the target and decoy peptides. This approach is termed "C-TDC" for combined TDC.

One apparent drawback of the C-TDC protocol is that the reported list of identified spectra contains a mixture of target and decoy peptides. In practice, of course, the user is typically interested only in the spectra that match a target peptide. Accordingly, the *target-only* variant of target-decoy competition (T-TDC) eliminates decoy identifications from the reported list and adjusts the FDR estimate accordingly [43, 42]. The FDR estimate is simply the number of decoys divided by the number of targets. Hence, for a fixed score threshold, the T-TDC protocol yields the same number of target identifications as C-TDC but a lower estimated FDR.

Unfortunately, the target-decoy competition that is the basis for both of these methods leads to two closely related problems. First, the competition occasionally eliminates a high-scoring target PSM because the corresponding decoy PSM happened to achieve an even higher score. Second, when using randomly generated decoy peptides, the TDC method exhibits an undesirable variability because the filtered target PSMs differ each time the procedure is run. Note that the use of reversed, rather than shuffled, decoy peptides simply hides this problem by arbitrarily fixing the decoys and the corresponding filtered peptides.

To avoid randomly discarding a small proportion of the high-scoring target PSMs, Käll et al. proposed an alternative method, which we call "separated target-decoy search" (STDS), in which the decoy PSMs are used separately to estimate the FDR among the target PSMs [37]. In STDS, all target PSMs above a specified threshold are reported to the user. A second, more sophisticated approach proposed by Käll et al., which we call "STDS-PIT," involves estimating one additional parameter, the "percentage of incorrect targets" (PIT), from the data. In STDS-PIT, the final FDR estimate is the STDS estimate multiplied by the PIT. However, the inclusion of this parameter is problematic, because the STDS methods estimate the significance of each target PSM using the set of all decoy PSMs, their use should be restricted to search engines that use fairly well calibrated scores [4].

We showed that the two protocols based on target-decoy competition are asymptotically accurate in estimating the FDR within their respective lists of PSMs (although for C-TDC that list

is not one we are typically interested in). On the other hand, the STDS procedure is conservative (overestimating the true FDR) and the STDS-PIT method is liberal (underestimating the true FDR). Consequently, motivated by these observations and by our desire for a method that avoids the drawbacks of target-decoy competition, we designed a *mixture-maximum* (mix-max) FDR estimation procedure that reports all sufficiently high scoring target PSMs while consistently estimating the FDR. The mix-max approach separately estimates the number of false discoveries due to foreign spectra and due to native spectra. The first part essentially follows the STDS-PIT approach; i.e., like STDS-PIT, mix-max estimates properties of the null distribution from the decoy set. The second is a bit more involved and requires estimating the distribution of W_i for a native spectrum. We define the resulting mix-max FDR estimation as

$$\frac{\widehat{\pi_0} \cdot \sum_{j=1}^{n_\Sigma} 1_{z_j>T} + (1-\widehat{\pi_0}) \cdot \sum_{z_j>T} \left[\frac{\sum_k 1_{w_k \le z_j}}{(1-\widehat{\pi_0}) \cdot \sum_k 1_{z_k \le z_j}} - \frac{\widehat{\pi_0}}{1-\widehat{\pi_0}} \right]_{[0,1]}}{\sum_i 1_{w_i>T}}.$$

where $1_{z_i>T}$, $1_{w_i>T}$, $1_{w_k \le z_j}$, $1_{z_k \le z_j}$ are 1 or 0, depending on whether the corresponding inequality holds, and where $\widehat{\pi_0}$ is the estimated proportion of foreign spectra, T is the score threshold, w_i and z_j refer to the observed target and decoy PSM scores and $[x]_{[0,1]} := \max\{0, \min\{1, x\}\}$ ensures that x remains an acceptable probability value. In the equation, the numerator is the sum of two terms, corresponding to the estimated number of false positives due to foreign and native spectra, respectively, and the denominator is the observed number of accepted (target) PSMs.

These results were published in the article entitled

> **Mix-Max: an improved false discovery rate estimation procedure for shotgun proteomics** written by Uri Keich, Attila Kertész-Farkas, William Stafford Noble and which appeared in **Journal of Proteome Research**, 2015, 14(8) 3148–3161

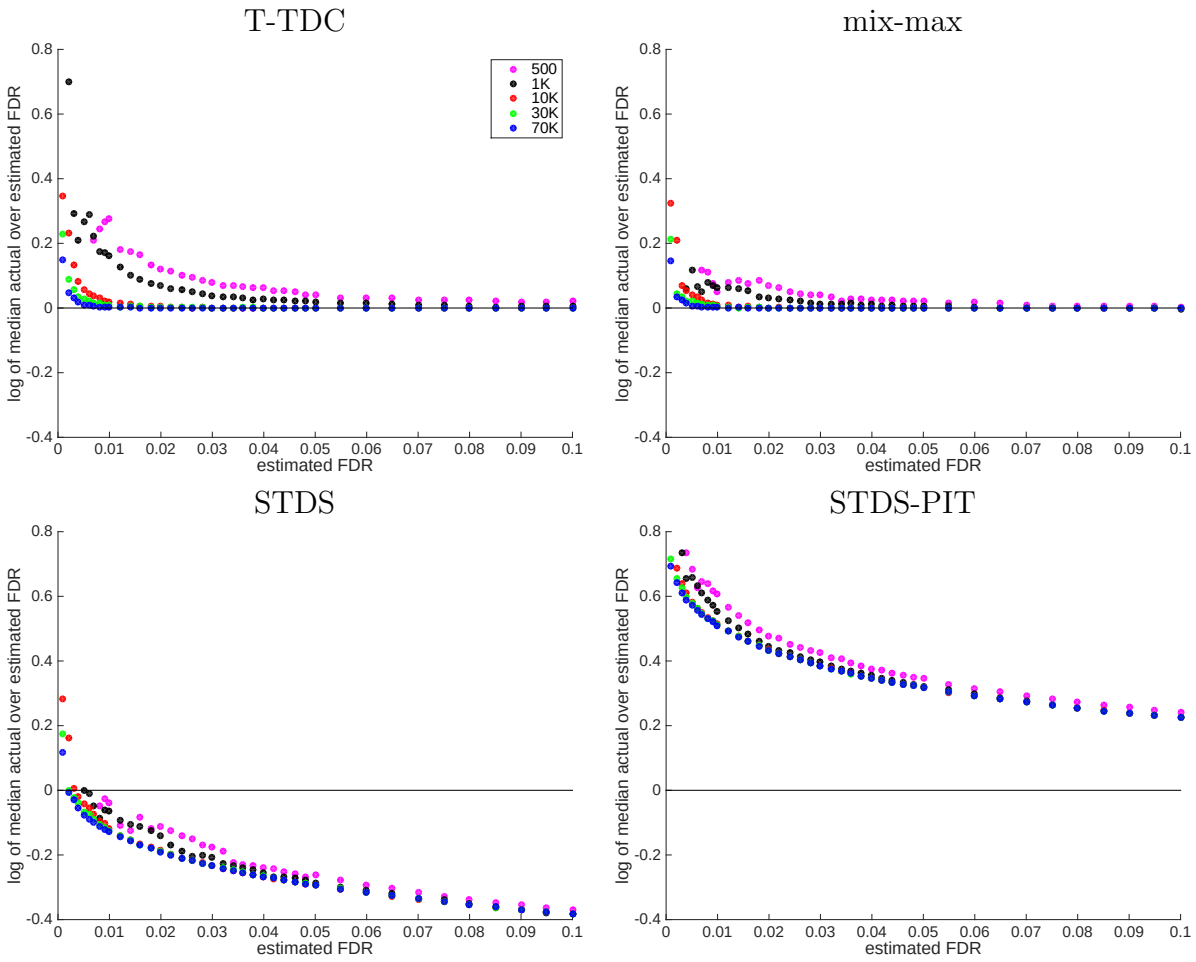

Figure 30: **Accuracy of estimated FDR (mixture model).** Each panel plots, as a function of estimated FDR, the logarithm of the median ratio between the actual FDR and the nominal one (so a value of 0 means perfect median estimation). The data was generated using the normal mixture model (see Methods), and the number of spectra increased from 500 to 70000 keeping the native spectra rate at 0.5. The medians are calculated at each FDR value with respect to 10K random draws of both native and foreign spectra. Because all the plots are on the same scale, it is easy to see that STDS overestimates the true FDR while STDS-PIT underestimates it. Both T-TDC and mix-max become increasingly more accurate as the spectrum set and/or the nominal FDR level is becomes larger, but mix-max seems slightly more accurate. In the case of both T-TDC and mix-max and small spectrum sets (500 and 1000) the median estimated FDR jumps from 0 to a number greater than 0.001; hence, the logarithm of the ratio to the nominal FDR is not defined for some small nominal FDR values. When the average separation between the correct PSM scores and the false PSM scores is further increased we noted similar results albeit STDS-PIT suffers a reduced bias whereas the opposite holds for STDS.

6.3 Modification searching

6.3.1 The PTMSearch method

The PTMSearch method is method for untargeted PTM identification; that is, it employs a large collection of known modification (PTM DB). The PTM DB typically contains the specifications of around 500-1000 known and curated PTMs. The PTMSearch method generates all possible modifications of a peptide sequence using the modifications from PTM DB and stores these sequences in a prefix tree T. For a peptide sequence $p_1 \dots p_n$ of length n, a prefix tree structure can be built wherein the nodes at the i-th level of the tree are labeled with the amino acid p_i and each branch represents different modifications on p_i taken from a PTMDB. The tree node at the level i is a structure $v = \langle z, b, y, m, c \rangle$, where z is a score of the node depending on the matching peaks of the fragment ions processed so far, c is the number and m is the sum of the mass of the acquired modifications in the sequence $p_1 \dots p_i$. Finally, the variables b and y store the masses (m/z) of the b and y fragment ions that correspond to the $p_1 \dots p_i$ and $p_{i+1} \dots p_n$ fragment ions respectively. This is illustrated in Figure 31. The PTMSearch algorithm relies on a Greedy Tree Traversal (GTT) approach, a variant of an A^* algorithm, to find the best modified peptide sequence in the tree which fits the experimental spectra the most. GTT employs a priority double ended queue Q, it inserts a node to queue Q when it is visited at the first time, deletes when all of its children have been visited, and continues the search from the node with the highest score in the Q. When the size of Q exceeds a certain limit Q_T, the node with the lowest score is deleted along with the corresponding subtree. This algorithm is described by the pseudocode in 4.

Algorithm 4: PTMSearch

```
Input  : Tree T
Output: Best goal (or NULL if there is no goal leaf)
  create an empty priority Q
  put(Q,root)
  while Q is not empty do
    v = front(Q)
    create a non-visited child v_j of node v;
    if all children of v has been visited then
      pop_first(Q);
    end if
    if v_j is a goal leaf then
      update best goal;
    else
      put(Q,v_j);
    end if
    if size(Q) > Q_T then
      pop_last(Q);
    end if
  end while
  Return best goal (or NULL if there is no goal leaf);
```

It may happen that the true goal leaf is eliminated from the tree by deleting a node from Q in the last if statement. We give an estimate about the probability of eliminating the true goal $P(\epsilon)$ under the following assumptions: The probability of a node $v \in T$ matches to a peak by chance (i.e. the score of a child of v is increased due to one of the fragment ions match with a noise peak) is $p = 2 * m * \delta / PM(s)$, where m is the number of the peaks in q and $PM(s)$ is the precursor mass of the spectrum s. We assume that the peaks are evenly and independently distributed in the experimental spectrum. Let p_e be the probability of that a (non-noise) peak is missing from the spectrum independently from other peaks (because either it was not observed or was filtered out in a preprocessing step).

Theorem 1. *Using the assumptions and parameters above the probability $P(\epsilon)$ of eliminating the true goal from the search space is $P(\epsilon) = N_L \cdot p \cdot (K + \sum_{j=1}^{H} p_e^j)$, where $H = Q_T/(N_L)^K$, $Q_T > M$ is the bound of the size of the queue Q, and K is the limit of the PTMs on a peptide to be identified, N_L is the expected number of modification per amino acid, and M is the maximal number of modifications which can modify one amino acid.*

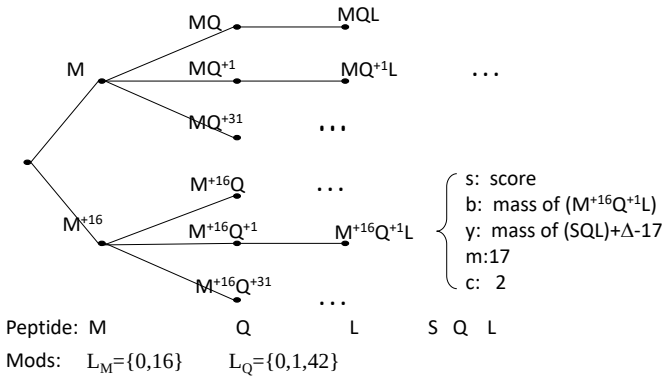

Figure 31: Illustration of a prefix tree of a peptide MQLSQL and its modified variants.

The results were published in the article entitled

PTMSearch: A Greedy Tree Traversal Algorithm for Finding Protein Post-Translational Modifications in Tandem Mass Spectra written by Attila Kertész-Farkas Beáta Reiz, Michael Myers, Sándor Pongor and which appeared in **European Conference on Machine Learning**, 2011, 29(7) 925–932

6.3.2 The PTMTreeSearch method

The PTMTreeSearch method is a further development of the PTMSearch method described in the previous section. While PTMSearch uses a GTT method, PTMTreeSearch traverses the whole tree T but it rather employs several tree pruning rules based on chemically and biologically plausible, i.e. when a branch in the tree T would lead to an likely incorrect annotation.

PTMTreeSearch uses two-stage search strategy. The first round uses strong tree pruning rules and serves to identify a smaller set of likely PTM types, while the second round uses a more loose

tree pruning rules to identify all possible occurrences of this restricted set of PTM types. This approach is based on the assumption that each modification type present in a sample has to be found in at least one good quality spectrum.

Evaluating our experimental results,we found out that the greedy approach identifies 36% more modification types, but 5% fewer peptides, 18% fewer modified peptides and takes three and half times longer than the current version of PTMTreeSearch. By benchmarking the performance of the PTMTreeSearch at 1% FDR, we found that PTMTreeSearch identifies 73% more and misses 13% less peptides on average in comparison against the state-of-the-art PTM identification methods, InsPecT, MODi and SIMS.

The results were published in the article entitled

> **PTMTreeSearch: a Novel Two-Stage Tree Search Algorithm with Pruning Rules for the Identification** written by Attila Kertész-Farkas, Beáta Reiz, Roberto Vera, Michael P. Myers, Sándor Pongor and which appeared in **Bioinformatics**, 2014, 30(2), 234–241

6.4 Spectrum filtering methods

6.4.1 The Precursor mass dependent filtering method

We have reviewed the spectrum filtering methods and their usage with score functions in Section 2.2.2. In our approach the number of positive peaks is considered to be a function of the number of the peptide bonds in a peptide. The number of peptide bonds can be approximately calculated by dividing the precursor mass, MH+, with the average molecular mass of the amino acid residues, AAM. If a mass spectrometer generates X fragment ions per peptide bonds, then estimated peak number (EPN) can be approximated by the following formula:

$$EPN(X) = X \cdot \left(\frac{MH^+}{AAM} - 1 \right), \tag{34}$$

where AAM denotes the average amino acid mass, approximately 120 Da. AAM is left as a variable, in order to take into account post translational modifications, which can change this value. The bracket term corresponds to the average number of peptide bonds for a precursor mass MH+. For instance, it is known that gas phase fragmentation can produce seven regular fragment ions for every peptide bond within a peptide. Then EPN(7) will approximate the maximum number of the positive peaks in the spectrum among the noise peaks produced by CID fragmentation. If only b and y fragment ions are expected, then the parameter X should be set to around 2.

The experimental results show that the mass-dependent filtering is more efficient than the other two filtering methods, and raw, unfiltered data provide the lowest number of correctly annotated spectra. The crossing point of the false discovery rate (FDR) line within the ROC curves can be used as an estimate of the correctly identified spectra at a given level of FDR. According to this estimate, when the performance of X!Tandem is tested at level of FDR= 0.2%, mass dependent filtering gives 10-15% more spectra the other two filtering techniques, and about 25% more than found with the raw data. In our opinion, the reason of this improvement is that both the Top-N and the Noise filter filters apply uniform filtering criteria for small and large peptides, while our mass dependent filter adjusts to the weight of the peptide molecule.

The results were published in the article entitled

> **Precursor Mass Dependent filtering of Mass Spectra for Proteomics Analysis** written by Beáta Reiz, Michael Myers, Sandor Pongor, Attila Kertész-Farkas* and which appeared in **Protein and Peptide Letters**, 2014, 21(8) 858–863

6.4.2 The Chemical rule-based filtering method

The mass dependent filtering has been developed further in order to obtain a more advanced filtering method in order to improve the spectrum annotations at any FDR level. More specifically, the Chemical Rule-based Filtering (CRB) seeks to retain (i) high-intensity peaks that are trusted without further conditions, and (ii) low-intensity peaks that are related to one of the high-intensity peaks according to any of the following three rules:

- Mass complementation rule: the sum of the masses of a pair of the high- and low-intensity peaks that add up to the precursor mass $MH^+ + 1$ (e.g. b–y pairs);
- Amino acid mass distance rule: the mass difference of two peaks equals to one of the known—native or modified— amino acid masses ("amino acid neighbors"); and
- Amino acid mass complementation rule: the sum of two peaks and the precursor mass MH^+ differs by one amino acid mass (e.g. a b-ion and the amino acid neighbor of its y-pair).

Certain types of instruments may not produce equally intense b–y (a–x or c–z) peak pairs, for instance, triple quadrupole or quadrupole-time-of-flight instruments tend to produce only either b- or y-ions, thus only one member of the ion pair is visible. In this case, CRF keeps such peaks if they are either intense enough to be considered high-intensity peaks or are separated by a mass of an amino acid from high-intensity peaks and hence pass the amino acid mass distance rule. Furthermore, when amino acids lose ammonia (NH3) or water (H2O) on collision-induced dissociation (CID) fragmentation under certain circumstances, or when some amino acids carry PTMs, CRF keeps these peaks if they are intense enough or they have a high-intensity amino acid neighbor and thus pass the first or the second rule. Peaks that do not have amino acid neighbors, such as internal fragment ions, might be retained if they are high-intensity peaks. The high- and low-intensity peaks are defined via user specified thresholds. As an approximation, the high- and the low-intensity peaks are defined as the percentage of the estimated maximum peak number (EMP). Assuming that under the conditions of CID, seven ions can be produced for every peptide bond, the number of peptide bonds can thus be roughly calculated by dividing the precursor mass by the average mass of amino acids, so EMP can be calculated as

$$EMP(X) = \frac{7 \cdot MH^+}{AAM}, \tag{35}$$

where AAM denotes the average amino acid mass, which is calculated from a table of amino acid masses, plus the masses of any modifications that are specified. For instance, by specifying 30% as the upper threshold and 70% as the lower threshold, we select 30% of the EMP as 'high-intensity' peaks and the next 30–70% as 'low-intensity' ones. Peaks in the 71–100% interval are discarded. EMP is a rough estimate because it does not contain irregular fragmentation events. However, we found it useful because it does not penalize either small or large peptides.

By analyzing our experimental results, we found that CRF improves spectrum annotation by 15-25% at the same FDR level, and provides an approx. 75% compression of the data.

The results were published in the article entitled

> **Chemical rule-based filtering of MS/MS spectra** written by Beáta Reiz, Attila Kertész-Farka Michael Myers, Sandor Pongor and which appeared in **Bioinformatics**, 2013, 29(7) 925–932

6.5 The Crux toolkit

The Crux mass spectrometry analysis toolkit is an open source project that aims to provide users with a cross-platform suite of analysis tools for interpreting protein mass spectrometry data. The toolkit includes several search engines for both standard and cross-linked database search, as well as a variety of pre- and post-processing engines for assigning high-resolution precursor masses to spectra, assigning statistical confidence estimates to spectra, peptides and proteins, and performing label free quantification. Crux comes pre-complied for the Linux, Windows and MacOS operating systems. It is implemented as a single program that offers a wide variety of commands and handles various data formats.

The toolkit was published in the article entitled

> **Crux: rapid open source protein tandem mass spectrometry analysis** written by Sean McIlwain, Kaipo Tamura, Attila Kertész-Farkas, Charles Grant, ... (+7), William Stafford Noble and which appeared in **Journal of Proteome Research** in 2014, 13(10):4488–4491

6.6 Review papers

6.6.1 Overview on database-searching approach

A review article on the common stages of the most database-searching-based spectrum annotation tools have been published in the article entitled

> **Database searching in mass spectrometry based proteomics** written by Attila Kertész-Farkas, Beáta Reiz, Michael Myers, Sándor Pongor and which appeared in **Current Bioinformatics**, 2012, 7(2) 221–230

6.6.2 Review on spectrum data filtering methods

A review article on the common mass spectrometry data filtering methods have been published in the article entitled

> **Database searching in mass spectrometry based proteomics** written by Beáta Reiz, Attila Kertész-Farkas, Michael Myers, Sándor Pongor and which appeared in **Current Bioinformatics**, 2012, 7(2) 212–220

7 Conclusions

The computational mass spectrometry field has turned into a mature scientific field with well performing standard models providing high baselines. The computational methods have been constantly developed for around three decades and a deep knowledge has been accumulated in the scientific literature about the nature of the score statistics in any tandem mass spectrometry data analysis software tools.

However, the spectrum annotation and analysis problem is not a solved problem yet. Novel and better instruments with superior accuracy are being developed for novel specific scientific problems as well as novel methodologies emerging such data-independent-acquisition (DIA), top-down protein identification, immunopeptidomics. These technologies require novel and different computational programs; therefore, the development of computational methods for specific cases will continue.

8 List of abbreviations and conventions

CAM	Carbamidomethylation
CP	Candidate peptides
Da	Dalton
DB	Database
ESNL	Experiment-specific null distribution
FDR	False discovery rate
HRFS	High-resolution fragmentation settings
IP	Inner product
LRFS	Low-resolution fragmentation settings
MPA	Mass of the precursor ion
PPM	Parts per Million
PSM	Peptide-spectrum match
PTM	Post-translational modification
SFIP	Secondary fragmentation ion product
SPC	Shared peak count
SSNL	Spectrum-specific Null-distribution
TDA	Target-decoy approach
TMT	Tandem mass tag
XCORR	Cross correlation scoring

References

[1] Ruedi Aebersold and Matthias Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198, 2003.

[2] Attila Kertész-Farkas, Beáta Reiz, Michael P Myers, and Sándor Pongor. Database searching in mass spectrometry based proteomics. *Current Bioinformatics*, 7(2):221–230, 2012.

[3] 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.

[4] Uri Keich and William Stafford Noble. On the importance of well-calibrated scores for identifying shotgun proteomics spectra. *Journal of Proteome Research*, 14(2):1147–1160, 2014.

[5] Pavel Sulimov and Attila Kertész-Farkas. Tailor: universal, rapid, non-parametric score calibration method for database search-based peptide identification in shotgun proteomics. *bioRxiv preprint bioRxiv:10.1101/831776*, 2019.

[6] William S Noble. How does multiple testing correction work? *Nature biotechnology*, 27(12):1135, 2009.

[7] David N Perkins, Darryl JC Pappin, David M Creasy, and John S Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *ELECTROPHORESIS: An International Journal*, 20(18):3551–3567, 1999.

[8] David Fenyö and Ronald C Beavis. A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Analytical chemistry*, 75(4):768–774, 2003.

[9] Sean McIlwain, Kaipo Tamura, Attila Kertesz-Farkas, Charles E Grant, Benjamin Diament, Barbara Frewen, J Jeffry Howbert, Michael R Hoopmann, Lukas Käll, Jimmy K Eng, et al. Crux: rapid open source protein tandem mass spectrometry analysis. *Journal of Proteome Research*, 13(10):4488–4491, 2014.

[10] Jimmy K Eng, Tahmina A Jahan, and Michael R Hoopmann. Comet: an open-source ms/ms sequence database search tool. *Proteomics*, 13(1):22–24, 2013.

[11] Jurgen Cox, Nadin Neuhauser, Annette Michalski, Richard A Scheltema, Jesper V Olsen, and Matthias Mann. Andromeda: a peptide search engine integrated into the maxquant environment. *Journal of Proteome Research*, 10(4):1794–1805, 2011.

[12] Jürgen Cox and Matthias Mann. Maxquant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification. *Nature biotechnology*, 26(12):1367–1372, 2008.

[13] Viktoria Dorfer, Peter Pichler, Thomas Stranzl, Johannes Stadlmann, Thomas Taus, Stephan Winkler, and Karl Mechtler. Ms amanda, a universal identification algorithm optimized for high accuracy tandem mass spectra. *Journal of Proteome Research*, 13(8):3679–3684, 2014.

[14] Craig D Wenger and Joshua J Coon. A proteomics search algorithm specifically designed for high-resolution tandem mass spectra. *Journal of Proteome Research*, 12(3):1377–1386, 2013.

[15] William Stafford Noble and Michael J MacCoss. Computational and statistical analysis of protein mass spectrometry data. *PLoS computational biology*, 8(1):e1002296, 2012.

[16] Viktor Granholm and Lukas Käll. Quality assessments of peptide–spectrum matches in shotgun proteomics. *Proteomics*, 11(6):1086–1093, 2011.

[17] Sangtae Kim and Pavel A Pevzner. Ms-gf+ makes progress towards a universal database search tool for proteomics. *Nature communications*, 5:5277, 2014.

[18] Bernhard Y Renard, Marc Kirchner, Flavio Monigatti, Alexander R Ivanov, Juri Rappsilber, Dominic Winter, Judith AJ Steen, Fred A Hamprecht, and Hanno Steen. When less can yield more–computational preprocessing of ms/ms spectra for peptide identification. *Proteomics*, 9(21):4978–4984, 2009.

[19] Yong J Kil, Christopher Becker, Wendy Sandoval, David Goldberg, and Marshall Bern. Preview: a program for surveying shotgun proteomics tandem mass spectrometry data. *Analytical chemistry*, 83(13):5259–5267, 2011.

[20] 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, 2004.

[21] John R Yates, Jimmy K Eng, Ashley L McCormack, and David Schieltz. Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. *Analytical chemistry*, 67(8):1426–1436, 1995.

[22] Jimmy K Eng, Michael R Hoopmann, Tahmina A Jahan, Jarrett D Egertson, William S Noble, and Michael J MacCoss. A deeper look into comet—implementation and features. *Journal of the American Society for Mass Spectrometry*, 26(11):1865–1874, 2015.

[23] 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. *Journal of Proteome Research*, 10(5):2671–2678, 2011.

[24] 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, 2008.

[25] Sangtae Kim, Nikolai Mischerikow, Nuno Bandeira, J Daniel Navarro, Louis Wich, Shabaz Mohammed, Albert JR 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.

[26] Aaron A Klammer, Christopher Y Park, and William Stafford Noble. Statistical calibration of the sequest xcorr function. *Journal of Proteome Research*, 8(4):2106–2113, 2009.

[27] Victor Spirin, Alexander Shpunt, Jan Seebacher, Marc Gentzel, Andrej Shevchenko, Steven Gygi, and Shamil Sunyaev. Assigning spectrum-specific p-values to protein identifications by mass spectrometry. *Bioinformatics*, 27(8):1128–1134, 2011.

[28] Yulia Danilova, Anastasia Voronkova, Pavel Sulimov, and Attila Kertesz-Farkas. Bias in false discovery rate estimation in mass-spectrometry-based peptide identification. *Journal of Proteome Research*, 18(5):2354–2358, 2019.

[29] Rovshan G Sadygov and John R Yates. A hypergeometric probability model for protein identification and validation using tandem mass spectral data and protein sequence databases. *Analytical chemistry*, 75(15):3792–3798, 2003.

[30] J Jeffry Howbert and William Stafford Noble. Computing exact p-values for a cross-correlation shotgun proteomics score function. *Molecular & Cellular Proteomics*, 13(9):2467–2479, 2014.

[31] Andy Lin, J Jeffry Howbert, and William Stafford Noble. Combining high-resolution and exact calibration to boost statistical power: A well-calibrated score function for high-resolution ms2 data. *Journal of Proteome Research*, 17(11):3644–3656, 2018.

[32] 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, 2007.

[33] Pavel Sulimov, Anastasia Voronkova, and Attila Kertész-Farkas. Annotation of tandem mass spectrometry data using stochastic neural networks in shotgun proteomics. *Bioinformatics*, 36(12):3781–3787, 2020.

[34] Siegfried Gessulat, Tobias Schmidt, Daniel Paul Zolg, Patroklos Samaras, Karsten Schnatbaum, Johannes Zerweck, Tobias Knaute, Julia Rechenberger, Bernard Delanghe, Andreas Huhmer, et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature methods*, 16(6):509, 2019.

[35] Ralf Gabriels, Lennart Martens, and Sven Degroeve. Updated ms^{2}pip web server delivers fast and accurate ms^2 peak intensity prediction for multiple fragmentation methods, instruments and labeling techniques. *Nucleic acids research*, 47(W1):W295–W299, 2019.

[36] Bradley Efron. Microarrays, empirical bayes and the two-groups model. *Statistical science*, pages 1–22, 2008.

[37] Lukas Käll, John D Storey, Michael J MacCoss, and William Stafford Noble. Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *Journal of Proteome Research*, 7(01):29–34, 2007.

[38] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[39] Imre Csiszár. Information-type measures of difference of probability distributions and indirect observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

[40] John T Halloran, Jeff A Bilmes, and William S Noble. Learning peptide-spectrum alignment models for tandem mass spectrometry. *Conference on Uncertainty in Artificial Intelligence*, 30:320, 2014.

[41] Xin Huang, Lin Huang, Hong Peng, Ashu Guru, Weihua Xue, Sang Yong Hong, Miao Liu, Seema Sharma, Kai Fu, Adam P Caprez, et al. Isptm: an iterative search algorithm for systematic identification of post-translational modifications from complex proteome mixtures. *Journal of proteome research*, 12(9):3831–3842, 2013.

[42] 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, 2007.

[43] K. Jeong, S. Kim, and N. Bandeira. False discovery rates in spectral identification. *BMC Bioinformatics*, 13(Suppl. 16):S2, 2012.

9 Appendix A - Declarations of author contribution

Article #1

In the article entitled

> **Deep convolutional neural networks help scoring tandem mass spectrometry data in database-searching approaches** written by Polina Kudriavtseva, Matvey Kashkinov, Attila Kertész-Farkas and which appeared in **Journal of Proteome Research** in 2021, `https://doi.org/10.1021/acs.jproteome.1c00315`

The personal contributions of Attila Kertész-Farkas are the following:

- Conceived the idea of Slider.
- Designed the experiments.
- Wrote most part of the manuscript.

These are the main contributions to this article. Matvey Kashkinov implemented a prototype of the Slider method, which was a shallow convolutional neural network and trained it in an unsupervised manner. The results outperformed baselines but not by a large margin. Polina Kudriavtseva has implemented Slider by a deep convolutional neural network and trained it via supervised manner. These results have achieved outstanding performance.

Article #2

In the article entitled

> **Tailor: non-parametric and rapid score calibration method for database search-based peptide** written by Pavel Sulimov, Attila Kertész-Farkas and which appeared in **Journal of Proteome Research** in 2020, 18(5), 2354–2358

The personal contributions of Attila Kertész-Farkas are the following:

- Designed the Tailor method based on an novel empirical observation.
- Designed the experiments.
- Implemented Tailor in Crux toolkit.
- Wrote most part of the manuscript.

These are the main contributions to this article. Pavel Sulimov implemented a prototype of the Tailor and other early versions of the Tailor method and tested them.

Article #3

In the article entitled

> **Bias in false discovery rate estimation in mass-spectrometry-based peptide identification** written by Yulia Danilova, Anastasia Voronkova, Pavel Sulimov, Attila Kertész-Farkas and which appeared in **Journal of Proteome Research**, 2019, 18(5), 2354–2358

The personal contributions of Attila Kertész-Farkas are the following:

- Conceived the main idea that theoretical target and decoy peptides could be classified with machine leaning methods,
- Designed the experiments,
- Wrote the major part of the manuscript.

These are the main contributions to this article.

Article #4

In the article entitled

> **Tandem Mass Spectrum Identification via Cascaded Search** written by Attila Kertész-Farkas, Uri Keich, William Stafford Noble and which appeared in **Journal of Proteome Research**, 2015, 14(8), 3027–3038

The personal contributions of Attila Kertész-Farkas are the following:

- Implemented the Cascade method and the baseline methods such as Group-FDR,
- Performed the experiments and analyzed the data,
- Contributed article writing.

These are the main contributions to this article.

Article #5

In the article entitled

> **PTMTreeSearch: a Novel Two-Stage Tree Search Algorithm with Pruning Rules for the Identification** written by Attila Kertész-Farkas, Beáta Reiz, Roberto Vera, Michael P. Myers, Sándor Pongor and which appeared in **Bioinformatics**, 2014, 30(2), 234–241

The personal contributions of Attila Kertész-Farkas are the following:

- Conceived the PTMTreeSearch algorithm,
- Designed and performed the experiments,
- Wrote tha major part of the manuscript.

These are the main contributions to this article. Roberto Vera constructed a web engine for the PTMTreeSearch method. Beata Reiz, Sandor Pongor and Michael P. Myers provided background on mass spectrometry and proteomics.

Article #6

In the article entitled

> **PTMSearch: A Greedy Tree Traversal Algorithm for Finding Protein Post-Translational Modifications in Tandem Mass Spectra** written by Attila Kertész-Farkas Beáta Reiz, Michael Myers, Sándor Pongor and which appeared in **European Conference on Machine Learning**, 2011, 29(7) 925–932

The personal contributions of Attila Kertész-Farkas are the following:

- Conceived and implemented the PTMSearch algorithm,
- Designed and performed the experiments,
- Wrote major part of the manuscript.

These are the main contributions to this article.

Article #7

In the article entitled

> **Precursor Mass Dependent filtering of Mass Spectra for Proteomics Analysis** written by Beáta Reiz, Michael Myers, Sandor Pongor, Attila Kertész-Farkas* and which appeared in **Protein and Peptide Letters**, 2014, 21(8) 858–863

The personal contributions of Attila Kertész-Farkas are the following:

- Conceived the idea and implemented the method,
- Designed and performed the experiments,
- Wrote the manuscript.

These are the main contributions to this article.

Article #8

In the article entitled

> **Annotation of tandem mass spectrometry data using stochastic neural networks in shotgun proteomics** written by Pavel Sulimov, Anastasia Voronkova, Attila Kertész-Farkas and which appeared in **Bioinformatics** in 2020, 18(5), 2354–2358

The personal contributions of Attila Kertész-Farkas are the following:

- Conceived the main idea that Boltzmann machines could be used for spectrum identification,
- Planned research work and distributed work among co-authors,
- Wrote most part of the article.

Article #9

In the article entitled

> **Improved False Discovery Rate Estimation Procedure for Shotgun Proteomics** written by Uri Keich, Attila Kertész-Farkas, William Stafford Noble and which appeared in **Journal of Proteome Research**, 2015, 14(8) 3148–3161

The personal contribution of Attila Kertész-Farkas is the following:

- Implemented and tested the Mix-Max method in C++ to the Crux-toolkit.

Article #10

In the article entitled

> **Crux: rapid open source protein tandem mass spectrometry analysis** written by Sean McIlwain, Kaipo Tamura, Attila Kertész-Farkas, Charles Grant, ... (+7), William Stafford Noble and which appeared in **Journal of Proteome Research** in 2014, 13(10):4488–4491

The personal contributions of Attila Kertész-Farkas are the following:

- Implemented several methods to the Crux Toolkit,
- Fixed bugs in tide-search and tide-index which are modules of Crux toolkit.
- Contributed to the article writing.

Article #11

In the article entitled

> **Chemical rule-based filtering of MS/MS spectra** written by Beáta Reiz, Attila Kertész-Farka Michael Myers, Sandor Pongor and which appeared in **Bioinformatics**, 2013, 29(7) 925–932

The personal contributions of Attila Kertész-Farkas are the following:

- Designed the experiments,
- Wrote major part of the manuscript.

Article #12

In the article entitled

> **Guided Layer-wise Learning for Deep Models using Side Information** written by Pavel Sulimov, Elena Sukmanova, Roman Chereshnev, Attila Kertész-Farkas* and which appeared in **Communications in Computer and Information Science**, 2020, 1086: 50–61

The personal contributions of Attila Kertész-Farkas are the following:

- Conceived the idea,
- Designed the experiments,
- Wrote the manuscript.

These are the main contributions to this article.

Article #13

In the article entitled

Database searching in mass spectrometry based proteomics written by Attila Kertész-Farkas, Beáta Reiz, Michael Myers, Sándor Pongor and which appeared in **Current Bioinformatics**, 2012, 7(2) 221–230

The personal contributions of Attila Kertész-Farkas are the following:

- Planned and organized the article,
- Wrote the article.

These are the main contributions to this article.

Article #14

In the article entitled

Data preprocessing and filtering in mass spectrometry based proteomics written by Beáta Reiz, Attila Kertész-Farkas, Michael Myers, Sándor Pongor and which appeared in **Current Bioinformatics**, 2012, 7(2) 212–220

The personal contributions of Attila Kertész-Farkas are the following:

- Contributed to article writing.