National Research University Higher School of Economics

as a manuscript

Pavel Sulimov

LEARNING GENERATIVE PROBABILISTIC MODELS FOR MASS SPECTROMETRY DATA IDENTIFICATION

Ph.D. Dissertation Summary

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

Moscow — 2020

This work was prepared at National Research University Higher School of Economics.

Academic Supervisor:	Attila Kertész-Farkas, PhD Associate Professor, Faculty of Computer Science, National Research University Higher School of Economics

1 Introduction

Mass spectrometry is used to study and identify molecules in a mixture of samples. The spectrometer generates sort of fingerprints of molecules, called spectra, which are then subjected to identification or annotation of the original molecules which could have generated the given spectra.

Mass spectrometry has gained attention in various fields including molecular biology, forensic, pharmaceutical industry, medicine, etc. For instance, in environmental containment 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.

This thesis focuses on protein identification in a biological mixture from spectrum data obtained with tandem mass spectrometry.

1.1 The relevance of research

On one hand, tremendous data has been accumulated in biological data repositories due to sharp drop of the cost of data storage devices in the past few years. On the other hand, the development of computational devices in the past few years such as the graphical processing units (GPUs) allows researches to develop computationally expensive and data intensive methods in short time. As a consequence, several deep learning based methods have been published for spectrum data annotation recently, such as, MS2PIP [9], pDeep [64], Prosit [21], DeepNovo [48], just to name a few.

1.2 Aims and objectives of research

In general, the goal of my research project was to develop more accurate spectrum identification methods. I developed a novel method, called BoltzMatch, which is based on a stochastic neural network and can annotate spectrum data more accurately; moreover, in contrary to other deep, black-box models, BoltzMatch is interpretable. At the time of writing this thesis, BoltzMatch achieved the state-of-the-art performance in spectrum annotation. More specifically, the development of the BoltzMatch method involved the following tasks:

1. BoltzMatch required a novel score calibration method, termed Tailor, to ensure that spectrum annotation scores obtained with it are normalized and thus comparable.

2. The training of BoltzMatch required the development of a novel regularization method. This method was termed diversifying regularization. I showed that the regularization helps train arbitrary deep, stochastic neural networks as well.

3. I showed that machine learning methods may overfit and result in biased error estimation due to their high model capacity. I showed that BoltzMatch does not acquire bias during its training.

1.3 Importance of work

Incorrect spectrum annotation may lead experimental scientist and practitioners to misleading conclusions about their experiments and to inaccurate decision making; for instance, in selecting the right drug therapy. Therefore, it is important to develop reliable and accurate methods to annotate and identify spectrum, in fact, any types of data.

1.4 Publications

My PhD research work has resulted in 4 main articles, of which three of them have been published in Q1 journals. Ranking is based on Scopus and Web of Science.

First-tier publications.

1. Sulimov P., Voronkova A., Kertész-Farkas A. Annotation of tandem mass spectrometry data using stochastic neural networks in shotgun proteomics. **Bioinformatics, Q1 journal**, 2020, doi: https://doi.org/10.1093/bioinformatics/btaa206, 2020.

2. Sulimov P., Kertész-Farkas A. Tailor: A Nonparametric and Rapid Score Calibration Method for Database Search-Based Peptide Identification in Shotgun Proteomics. **Journal of Proteome Research, Q1 journal**, 2020, doi: https://doi.org/10.1021/acs.jproteome.9b00736.

3. Danilova Y., Voronkova A., Sulimov P., Kertész-Farkas A. Bias in False Discovery Rate Estimation in Mass-Spectrometry-Based Peptide Identification. **Journal of Proteome Research, Q1 journal**, 2019, doi: https://doi.org/10.1021/acs.jproteome.8b00991

Second-tier publications.

4. Sulimov P., Sukmanova E., Chereshnev R., Kertész-Farkas A. Guided Layer-Wise Learning for Deep Models Using Side Information. In: van der Aalst W. et al. (eds) Analysis of Images, Social Networks and Texts. AIST 2019. **Communications in Computer and Information Science, Q3 book series**, vol 1086. Springer, 2020, doi: https://doi.org/10.1007/978-3-030-39575-9_6

Reports at conferences and seminars.

5. Guided Layer-wise Learning for Deep Models using Side Information, 8th International Conference - Analysis of Images, Social networks and Texts, 17-19 July 2019, Kazan, Russia.

6. Generative probabilistic modelling of peptide-spectrum matching in tandem mass spectrometry, X International forum "Biotechnology: State Of The Art and Perspectives", 25-27 February 2019, Moscow, Russia.

7. Modification of Restricted Boltzmann Machines for peptide identification, Annual Interuniversity Scientific and Technical Conference of Students, Postgraduates and Young Specialists named after E.V. Armensky, MIEM HSE, 19 February 2018, Moscow, Russia.

8. High-dimensional Generative Probabilistic Models for Peptide-spectrum Matching in Tandem Mass Spectrometry, II Russian-French workshop "Big Data and Applications", 12-13 October 2017, Moscow, Russia.

1.5 The organization of the thesis

The thesis is organized as follows. The Chapter 2 provides a brief overview of mass-spectrometry, discusses biological sample preparation, data generation, database searching-based spectrum identification process and results validation. The Chapter 3 gives more detailed description of scoring functions properties and exposes why violation of them could lead to weak statistical power of scoring function. Finally, Chapter 4 summarizes the Authors results from the articles written during the whole research process.

2 Mass spectrometry

2.1 Biological sample preparation

In this thesis a biological sample is referred to a small aliquot containing a mixture of several proteins and the general aim is to identify the sequences of the protein in the sample [1, 47, 37, 49]. A sample can be, for instance, blood sample, or sample from cancer cell obtained with biopsy, or small molecules used by bacteria for communication with each other etc.

The proteins in the sample first are cleaved using digestion enzymes to produce smaller molecules called peptides for the following reasons:

1. to decrease the size of the molecules because instruments can not deal with large molecules and

2. to reduce the complexity of analysis.

Enzymes cleave proteins at certain positions. This procedure is called digestion or cleaving. Cleavage site depends on the enzyme and different enzymes may cleave at different sites.

The most popular enzyme is the trypsin which cleaves after Arginine (R) and Lysine (K) if it is not followed by Proline (P) from the C-terminal [53]. Enzymes may not cleave at certain sites if they are inaccessible because of protein structure.

2.2 Mass spectrometer and data generation

Mass spectrometer is the instrument which takes an digested sample aliquot as input and the instrument can take the sample in solid, liquid or gas phase. The instrument produces a set of spectra which can be considered as fingerprints of molecules to be identified with computer programs.

A general mass spectrometer consists of the following steps:

1. *Vaporization.* Transforms the sample to gas phase.

2. *Ionization.* This step charges positively the molecules, charged molecule (cation) is called precursor (molecular) ion. Mainly, molecules are charged either by removing electron(s) with minor change of mass/charge ratio or adding one or more protons which considerably changes the mass/charge ratio proportional to the mass of proton.

3. *Acceleration.* The ionized samples are accelerated using a magnetic field.

4. *Deflection.* In this step the propagation path of charged particles is deflected by a magnet. Heavier molecules are deflected less, lighter - more. At this stage charged particles are separated (sorted) according to their mass-per-charge ratio.

5. *Detection.* Charged particles are detected - nowadays it is done with resolution that is high enough to distinguish between isotopes; ions are counted.

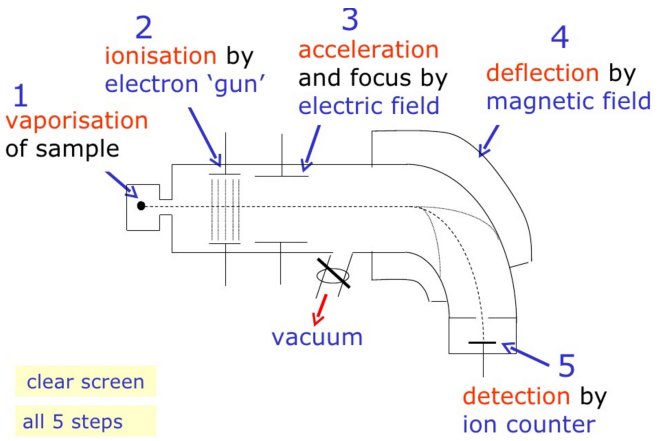

Figure 1: Pipeline of mass spectrometer work-flow [7].

The Figure 1 schema depicts a typical mass spectrometer.

One of the key differences between instruments is the type of ionization. For example, during electronic ionization [33, 28] electrons hit molecules and knock out one electron from the electron cloud, making molecules positively charged. At electrospray ionization (ESI) molecules are wrapped in micro droplets or coated by water or lipids, protecting the sample molecules. The molecular ions caught go through high pressure chambers, where the pressure heats up the sample and the liquid part evaporates. Only molecular ions remain and hits the detector. Electrospray ionization can be easily combined with high performance liquid chromatography: the flow of the chromatographic phase from the column can be sent directly to the capillary for electrospray. Thus, the mass spectrometer will determine the masses of molecules separated in the analytical column. This method is referred to as LC-MS (liquid chromatography-mass spectrometry) [51]. The identification of proteins in a complex solution using a combination of mass spectrometry and high performance liquid chromatography is called shotgun proteomics [2, 62, 19], which is currently a widely used method.

The *tandem mass spectrometry* (MS/MS) approach is illustrated in the Figure 2. Tandem mass spectrometers are used with "soft" ionization methods, which are methods that do not utilize electrons or chemical molecules. Thus, the first mass spectrometer analyses the whole molecular ions, and then, the molecular ions are fragmented further by collisions with inert gas molecules or laser radiation in the second mass spectrometer. In the end, each peptide is characterized by a set of masses and the abundance of the fragment ions, called a spectrum, along with the precursor ion.

The Figure 3 presents the hypothetical fragmentation of a given peptide *leuenkephalin*. In the peptide, the amide bond (CO-NH) is the most vulnerable and it is the most likely position where the peptide breaks into two parts called $b-$ion and $y-$ion. The peptide fragment containing the N-terminal is called the $b-$ion, while the other fragment containing the C-terminal is called the $y-$ion. The sum of the masses of the $b-$ and $y-$ions is equal to the mass of the original peptide ion. The distance between two neighbouring ions ($b-$ or $y-$ions) is equal to a mass of the amino acid which separates the two ions. Therefore, reading out the amino acids corresponding to the distances of the $b-$ion fragment series would tell the original peptide sequence (except the

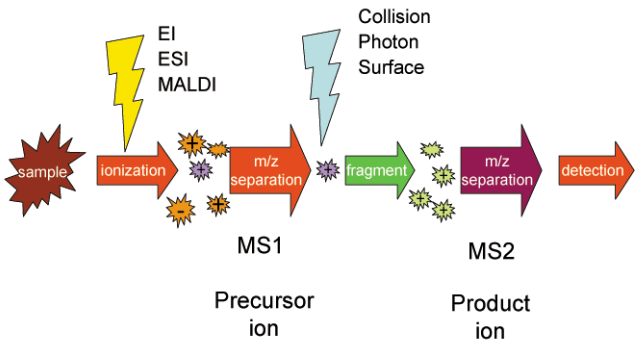

Figure 2: Schema of tandem mass spectrometry approach [5].

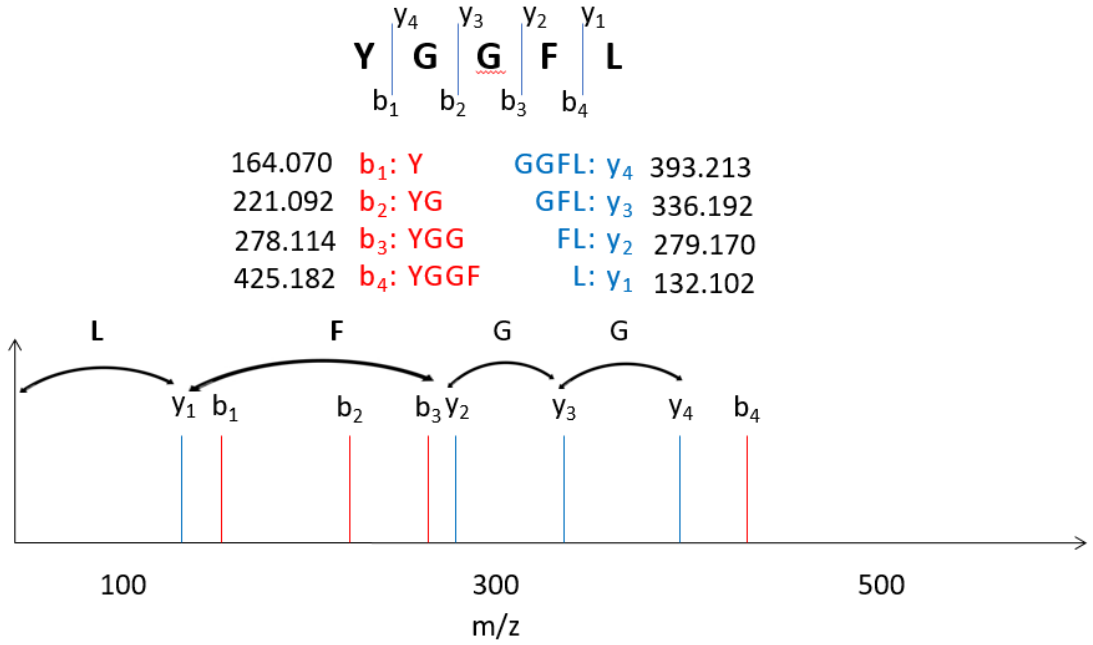

Figure 3: Theoretical fragmentation to b- and y-ions of peptide **YGGFL**. Fragments with masses are provided above the spectra. Double arrows with captions illustrate that the distance between two subsequent fragment ion is equal to the mass of corresponding amino acid.

last amino acid), while reading out the amino acids corresponding to the distances of the $y-$ion fragment series would tell the original peptide sequence in reverse order (except the first amino acid).

The Figure 4 shows a graphical illustration of a spectrum most likely to be generated by a peptide **HEEIDLVSLEEFYK**. The x-axis denotes the mass-to-charge ration m/z (Th). Mass is measured in Daltons (Da). Low-resolution mass spectrometers accuracy is within 1 Da, high-resolution mass spectrometers accuracy is up to 0.02 Da, which means it can make a difference between the 1/50 of the mass of a proton. The values of y-axis measure the abundance of the observed fragment ions.

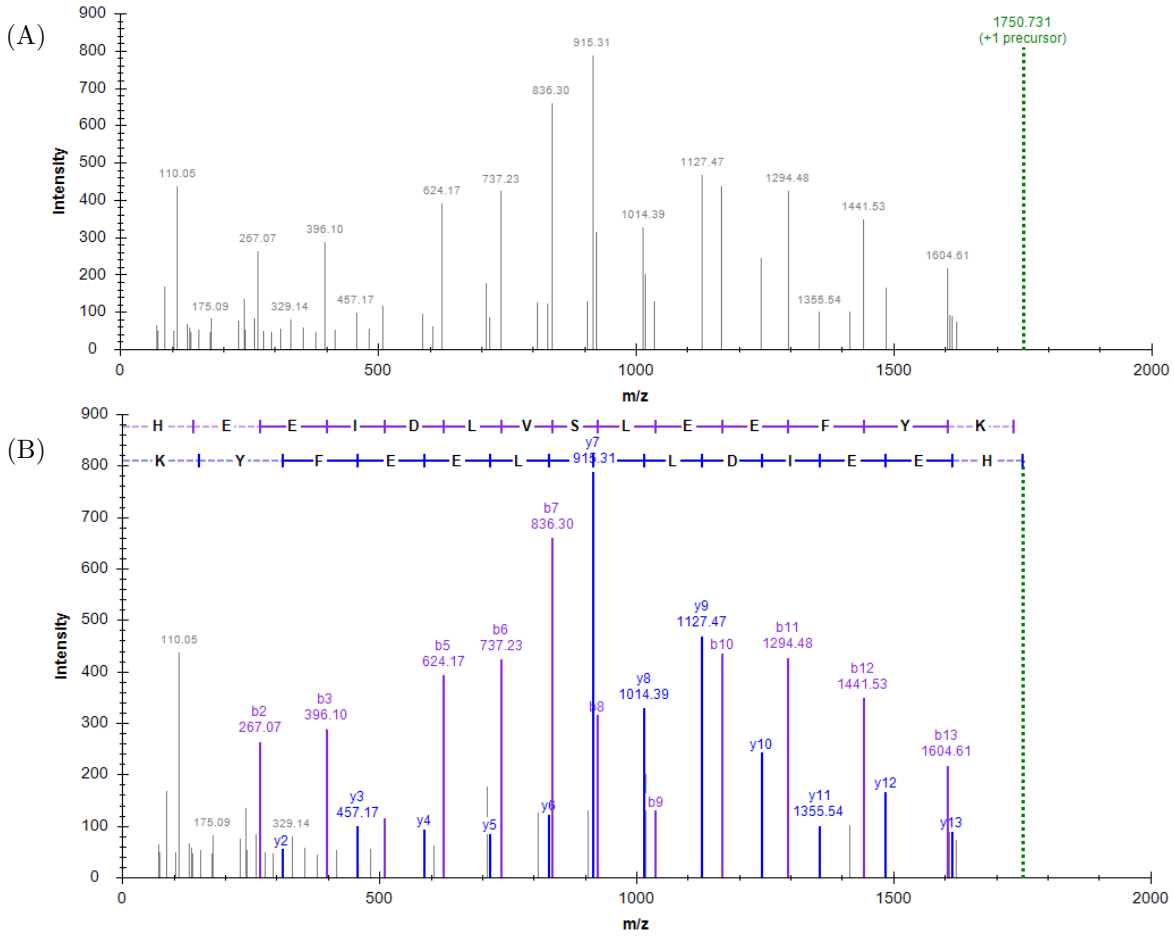

Figure 4: Mass spectrum (A) for **HEEIDLVSLEEFYK** with possibly correct annotation (B).

2.3 Database searching-based spectrum identification

In the database-searching approach, an observed spectra $s_i \in S$, where S denotes the set of spectra produced by the instrument from a single experiment, is annotated with the best scoring theoretical peptide, so-called peptide-spectrum match (PSM), found by iteratively matching s_i against a database of collected peptides, formally:

$$s_i \leftarrow \hat{h} = \arg\max_{h_j \in CP(s_i)} \phi(s_i, h_j). \tag{1}$$

This consists of three key elements: (1) the peptide database DB, (2) a selection of biologically/chemically plausible peptides, called candidate peptides ($CP(s_i) \subseteq DB$) for a given experimental spectrum s_i, and (3) the score function $\phi : S \times DB \rightarrow R$.

2.3.1 Peptide database

The peptide database is derived from protein sequence libraries, such as the UniProt, which are related to the experiment. For instance, if the biological samples are obtained from human, then the peptide database DB is to be constructed from human protein sequences.

The protein sequences are split at specific position according to the cleavage rules of the enzyme used in the sample preparation. This is also called *in silico* protein digestion [55]. In practice enzymes do not cleave properly or the biological samples undergo some modifications. The peptide database may need to handle these alterations. The typical options for peptide generation to handle them include (see the Table 1 for an example of trypsin cleavage with possible alternations):

- Missed cleavages: the number of the missed cleavage sites. This parameter usually ranges between 0-3.
- Non-enzymatic cleavage: whether one or both peptide terminals may result from non-enzymatic cleavage.
- Post-translational modifications: some molecules, for instance oxygen or phosphor, may attach to amino acid and consequently alter the masses of the fragment ions of the peptide (see Figure 5).

There are two main types of post-translational modification: static and variable [52]. Static modifications would be placed each time at definite amino acid - for example, Carbamidomethylation (CAM) is the result of reacting cysteine (C) residues with iodoacetamide [61]. This would cause change in weight of C amino acid with 57.02146 Da, and being "static" means that it would be applied to all cysteines in all peptides. Variable modification could occur in peptide, or could not - and when digesting proteins *in silico* the maximum number of variable modifications per peptide can be chosen. Example of variable modification could be oxidation on methionine (that would lead to increase in mass of methionine by 15.995 Da), or isobaric tandem mass tag (TMT)

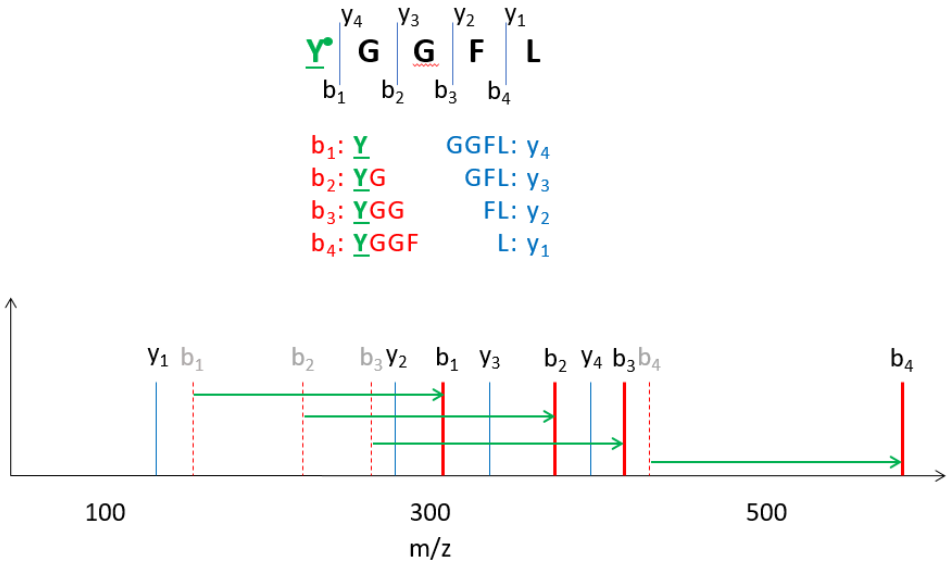

Figure 5: Tyrosine (Y) sulfation in **YGGFL** alters the weight of amino acids and the peptide that results peak shifts in the spectrum. One PTM shifts exactly half of all the peaks in peptide. In case modification happens not on first amino acid, some y-ions would be shifted as well.

labelling of lysine and N-terminal amino acids (N-terminal modification) leading to growth of mass by 229.16293 Da.

Original protein fragment
MEICRGLR|SHLITLLLFLFHSETICRPSGR|K|SSK|MQAFR|IWDVNQK|G...
Tryptic peptides
MEICRGLR, **MQAFR**, **IWDVNQK**, ...
Missed cleavages = 1
IWDVNQKG, **SGRK**, ...
Non-enzymatic peptides (spontaneous breakdown)
SHLITLLLF, **SGRKSS**, ...
Modified peptides
M(ox)EICRGLR, ...

Table 1: Fragment of protein from International Protein Index *IPI:IPI00000045.1—SWISS-PROT:P18510-1*. Vertical bars mark tryptic cleavage sites. The place of missed cleavage is highlighted with (red). The place where trypsin does not cleave because of Proline suppression rule is highlighted with (violet). **(ox)** indicates the oxidation on methionine (M).

2.3.2 Candidate peptides

The set of candidate peptides (CP) consists of peptides whose neutral mass of the precursor ion (MPA) is equal to the neutral mass of the precursor ion of s_i up to an instrument specific error tolerance. This restriction excludes peptides from the match, whose corresponding molecular masses are different from the mass of the precursor ion, thus they can not be correct eventually. In annotation, this restriction speeds up the searching and results in much less false annotations.

The CP is formally constructed as:

$$CP(s_i) = \{h_j : h_j \in DB, D(s_i, h_j, Z(s_i), \epsilon) \leq 0\} \tag{2}$$

where $Z(s_i)$ is the precursor charge state of s_i, D is a tolerance function, ϵ is *mass error tolerance* [54]. In practice, the tolerance function is usually defined in one of the following ways:

- In Dalton:
  $D(s_i, h_j, Z(s_i), \epsilon) = \left| \frac{MPA(h_j)}{Z(s_i)} - MPA(s_i) \right| - \epsilon$
- In PPM (Parts per Million):
  $D(s_i, h_j, Z(s_i), \epsilon) = \left(MPA(s_i) - \frac{MPA(h_j)}{Z(s_i)} \right)^2 - \left(MPA(s_i) * \frac{\epsilon}{1000000} \right)^2$.

2.3.3 Score functions

The score function measures a sort of similarity between the observed and the theoretical spectra, and higher score indicates a better spectrum match, which is based on counting the observed and theoretical peaks at the same position in the spectra s_i and h_j. However, due to measurement inaccuracy, peaks are considered matching up to a small instrument specific tolerance δ. For instance, if there is a peak at position p m/z in the observed spectrum and a peak at position q m/z in the theoretical spectrum, then these peaks are considered matching iff $|p - q| < \delta$. This approach is precise, albeit computationally exhaustive.

Another widely used approach involves spectrum discretization in which every spectrum is converted to a real-valued vector v, and peaks are placed in vector bins. A peak at position p m/z is placed in the bin $k = \left[\frac{p+(Z-1)*MP}{Z*res} + 1.0 - offset \right]$, and the value of the $v[k]$ is the intensity of the peak. Here, MP stands for the mass of the proton. If more than one peak would fall in the same bin, the value of $v[k]$ is usually the maximum of the intensities of the peaks which fall in the same bin k. Note that peak charges are not available for observed spectra and for them $Z = 1$; however, theoretical peptide peaks may be generated with multiple charge states, and Z may vary. For low resolution instruments it is common to use $res = 1.0005079, offset = 0.4$; for modern high resolution instruments: $res = 0.02, offset = 0.0$. Therefore, if the largest peak location considered is 2000 m/z, then the discretization results in a 1999-dimensional vector for low-resolution settings, and 100.000-dimensional vector for high resolution settings. For theoretical spectra the peak intensities are 1.0, resulting in binary vectors. The drawback of this approach is that peaks which are placed closer to each other than the predefined tolerance ($|p - q| < \delta$) may end up in adjacent vector bins.

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

Standard score functions are:

- Shared Peak Count (SPC) defined as

$$\text{SPC}(s_i, h_j) = \sum_{k=1}^{N} \mathbb{I}(s_i[k] \neq 0) \times \mathbb{I}(h_j[k] \neq 0), \tag{3}$$

where N is the number of bins. This approach does not take peak intensities into account.

- Inner product (IP):

$$\mathrm{IP}(s_i, h_j) = s_i^T h_j. \tag{4}$$

This function takes the intensities of the matching peaks into account.

- HyperScore, introduced by Fenyo et al. [17]:

$$\mathrm{HyperScore}(s_i, h_j) = \mathrm{IP}(q, t) \times N_b! \times N_y! \tag{5}$$

where $N_b!$ is the factorial of number of matched b-ions and $N_y!$ is the same but for matched y-ions. The idea behind is that, for example, 4 matching b-ions (or 4 matching y-ions) may indicate better spectrum peptide match than matching 2 b-ions and 2 y-ions.

- Cross correlation scoring (XCorr) was introduced with SEQUEST [15] as

$$\mathrm{XCorr}(s_i, h_j) = \mathrm{IP}(s_i, h_j) - \frac{1}{151} \sum_{\tau=-75}^{+75} \mathrm{IP}(s_i, h_j[\tau])), \tag{6}$$

The first part simply qualifies the match between the experimental and theoretical spectra using the inner product of the corresponding vectors The second part 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 a 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.

2.4 Search result validation

Database searching assigns one top-scoring peptide to every spectrum (see Table 2); however, this does not imply that the assignment is correct. The database-searching based peptide identification is (1) inexact meaning that the observed spectrum (1a) contains many unexplainable peaks which stem from the unusual fragmentation of the peptide or contaminating molecules, and (1b) lacks expected fragmentation ions, which fail to be observed in the mass spectrometer; and it is (2) incomplete, meaning that the peptide database is not complete and the "correct" peptide may not be included [49]. In practice, roughly the 40%-80% of the annotations can be incorrect. Consequently, spectrum assignment validation methods are essential.

Spectrum		Top-scored peptide	Score
s_1	$\leftarrow$	h_{1_j}	0.39
s_2	$\leftarrow$	h_{2_j}	0.97
s_3	$\leftarrow$	h_{3_j}	0.42
s_4	$\leftarrow$	h_{4_j}	1.13
...	...	...	...
s_n	$\leftarrow$	h_{n_j}	0.01

Table 2: List of top peptide-spectrum annotations.

2.4.1 Decoy peptides

The target-decoy approach (TDA) was introduced in 2007 [11] to validate spectrum annotations. This approach starts with generating fictitious peptide sequences, called *decoy* peptides and they are merged with the set of the original peptides, referred to as *target* peptides henceforth. There are four common ways to generate decoy peptides:

- protein reverse: in this approach the whole protein sequence is reversed and the decoy peptides are generated via *in silico* digestion as in the case of the target peptides;
- protein shuffle: here, the amino acids of the whole protein sequence are shuffled, and *in silico* digestion generates the decoy peptides. This approach generates more decoy peptides than target peptides because many protein sequences are homologs and hence share tryptic peptides. A protein-level shuffling scheme will shuffle all copies of these shared peptide multiple times, once for each occurrence in the proteome. The result is a decoy database that is larger than the target database, which in turn leads to conservative estimate of FDRs. [23];
- peptide reverse: the amino acids of the target peptides are reversed;
- peptide shuffle: the amino acids of the target peptides are shuffled.

Usually, the terminal amino acids are left in their places in the peptide-level decoy generation procedures. It is in need because the terminal amino acids are digestion enzyme specific and modifying these terminal amino acids would leak information about the type of the peptide, whether it is target or decoy. For instance, for the peptide ACCQPSTYK, the peptide reveres approach results in AYTSPQCCK, and the peptide shuffle approach can result in ATQPCSCYK.

Spectrum		Top-scoring peptide	Score
s_1	$\leftarrow$	$h^*_{1_j}$	0.39
s_2	$\leftarrow$	h_{2_j}	0.97
s_3	$\leftarrow$	$h^*_{3_j}$	0.42
s_4	$\leftarrow$	h_{4_j}	1.13
...	...	...	...
s_n	$\leftarrow$	h_{n_j}	0.01

Table 3: List of top peptide-spectrum annotations, decoy peptides are marked with (*).

Having generated the decoy peptides and pooled together with the list of target peptides, a standard database-searching step is carried out (see Table 3). Then the error in the spectrum annotation can be estimated based on the amount of the decoy peptides found in the search result under certain mild assumptions discussed in the next section.

2.4.2 FDR estimation

Assuming higher score means better annotation, PSMs can be ordered by their matching scores in descending order. At any particular score threshold, the PSMs above can be considered as trusted annotations also referred to as accepted PSMs. In practice, the ratio of the false discoveries is estimated among the accepted PSMs, where the number of the false discoveries (false positives) is estimated based on the number of the PSMs assigned to decoy peptides, called decoy PSMs. The false discovery rate (FDR) is calculated as follows:

$$\text{FDR} = E\left[\frac{FP}{R}\right], R > 0, \tag{7}$$

where FP indicates the number of the decoy PSMs and R denotes the number of the target PSMs, i.e. PSMs which are assigned to target peptides among the accepted PSMs. It has been shown by Levitsky et al. [44] that FDR estimation in Eq.7 leads to bias in estimation, thus +1 should be added to FP before dividing by R.

Accurate FDR control requires the following assumptions:

A-1) The set of target and decoy peptides must be distinct,

A-2) The target and decoy peptides should be generated independently,

A-3) The number of target and decoy peptides should be roughly equal sized; otherwise the FDR calculation must include a correction factor for the size.

A-4) Any spectra should be assigned to incorrect target or decoy peptide with equal likelihoods.

In practice, the score threshold is chosen so that the FDR level of the accepted PSMs is around a predefined α level.

2.4.3 Q-value calculation

The q-value of a PSM 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 1.

The example of q-values calculation using Algorithm 1 is provided in Table 4.

The example of q-value curves for different search methods is displayed in the Figure 6.

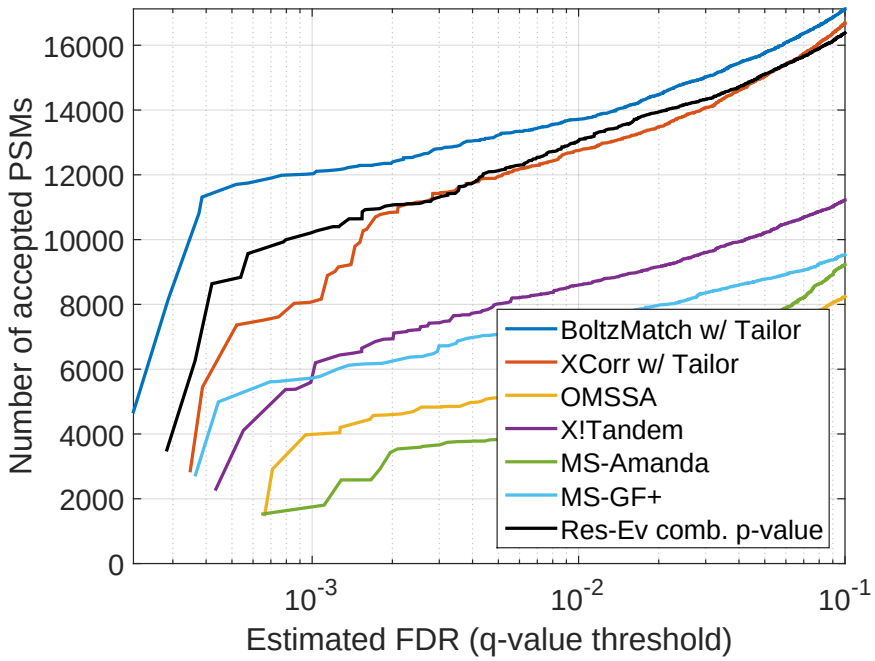

Figure 6: Spectrum annotation results with various search engines.X-axis denotes q-value thresholds, y-axis indicates the number of accepted PSMs at a given 1-value threshold. Different lines correspond to different methods; higher curves indicate better performance.

Algorithm 1: Q-value calculation

Input : List of best PSMs: $\phi(s_1, h_{1_j}) \, , \ldots , \, \phi(s_n, h_{n_j})$
Output: $Q_1 \ldots Q_n$
(Sort in decreasing order$(\phi(s_1, h_{1_j}) \, , \ldots , \, \phi(s_n, h_{n_j}))$)
$targets \leftarrow 0$
$decoys \leftarrow 1$
for $i = 1 \rightarrow n$ **do**
  if h_{i_j} is target peptide **then**
    $targets \leftarrow targets \, + \, 1$
  else
    $decoys \leftarrow decoys \, + \, 1$
  end if
  $FDR_i \leftarrow \frac{decoys}{targets}$
  if $FDR_i \, > \, 1$ **then**
    $FDR_i \leftarrow 1$
  end if
end for
$Q_1 \leftarrow FDR_1, \, \ldots , \, Q_n \leftarrow FDR_n$
for $i = n - 1 \rightarrow 1$ **do**
  if $Q_{i+1} < Q_i$ **then**
    $Q_i \leftarrow Q_{i+1}$
  end if
end for

Spectrum	Top scored peptide	Score	FDR	q-value
s_6	h_{6_j}	2.03	$0/1 = 0$	$= 0$
s_9	h_{9_j}	1.96	$0/2 = 0$	$= 0$
s_{11}	$h^*_{11_j}$	1.54	$1/2 = 0.5$	$= 0.16$
s_4	h_{4_j}	1.13	$1/3 = 0.33$	$= 0.16$
s_7	h_{7_j}	1.11	$1/4 = 0.25$	$= 0.16$
s_2	h_{2_j}	0.97	$1/5 = 0.2$	$= 0.16$
s_{10}	h_{10_j}	0.65	$1/6 = 0.16$	$= 0.16$
s_3	$h^*_{3_j}$	0.42	$2/6 = 0.33$	$= 0.33$
s_1	$h^*_{1_j}$	0.39	$3/6 = 0.5$	$= 0.38$
s_5	h_{5_j}	0.28	$3/7 = 0.43$	$= 0.38$
s_8	h_{8_j}	0.25	$3/8 = 0.38$	$= 0.38$

Table 4: Illustration of calculating FDRs and q-values.

2.5 De Novo and hybrid spectrum identification methods

Another approach to peptide identification is called *de novo* sequencing [46]. This approach is used for new proteomes, when protein sequence databases are not available for new species (new bacteria, etc.) The advantage of *de novo* sequencing is that it does not require a database of candidate proteins, but the disadvantage of this approach is that statistical approaches cannot be used to validate the identification. Database-searching-based approaches are used when protein sequence databases are available (for instance, for human tissues). De Novo methods uses biological and chemical rules to identify subsequences in the input spectrum s_i [4, 31].

Hybrid spectrum identification methods combines the de Novo with the database searching methods, in which the subsequences identified by the de Novo approach is used to filter $CP(s_i)$ by removing theoretical peptides h_j which do not contain any of the subsequences identified by the de Novo approach.

The results of this thesis are related to database searching-based approaches and I would not discuss de Novo methods deeper.

3 Scoring functions properties

Score functions are the workhorses in peptide identifications pipelines. Good score functions should be:

a) *discriminative*, meaning they separate correct PSMs from the incorrect ones,

b) *well-calibrated*, meaning they have a well defined and accurate semantics,

c) *unbiased*, meaning that they assign each spectrum to incorrect target or decoy peptide with equal likelihoods,

d) *universal*, meaning they work well for spectra generated using diverse configurations of MS instruments and experimental protocols [23, 40].

3.1 Discriminative property

The discriminative ability of a scoring functions means that the distribution of the scores corresponding to correct PSMs is well separated from the distribution of the incorrect PSMs; therefore, the correct PSMs can be separated from incorrect ones using simple thresholds.

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 [49]. 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 [6] 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; while the popular XCorr function of SEQUEST [15, 63] additionally incorporates signals from the flanking bins of the discretized spectrum vector [13], SFIP, and highly charged theoretical fragmentation ion masses depending on the charge state of the precursor ion. The XCorr is formalized as

$$\mathrm{XCorr}(s,h) = E(s,h) - Z(s,h) \tag{8}$$

for a discretized experimental spectrum s and a theoretical spectrum h, where E 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, and $Z(s,h)$ presents a correction factor, which is defined as $Z(s,h) = \frac{1}{151}\sum_{\tau=-75}^{+75} E(s,h[\tau])$, where in $h[\tau]$ all vector elements of h are shifted by τ steps [15, 57]. The weights can be arranged in a weight matrix W, providing the formalization $E(s,h) = s^T W h$, where T denotes vector transposition.

Several new score functions and database searching tools have also been introduced, including Mascot [50], HyperScore of X!Tandem[17], Morpheus [60], and MS Amanda [10]; however, these methods are based on manually constructed score functions and have resulted in only minor

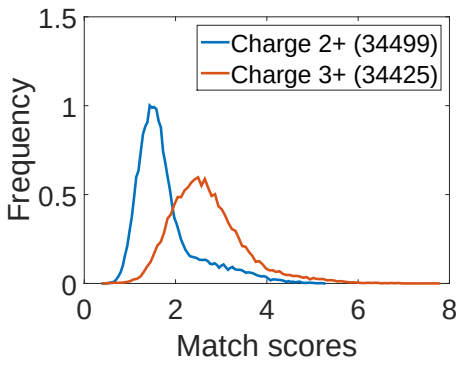

Figure 7: Distributions of top scoring PSMs obtained with simple match on Yeast data for doubly (blue) and triply (red) charged precursor ions.

improvements as compared with SEQUEST [38]. Recent studies focused rather on score calibration to provide a well-defined accurate semantics so that the spectrum annotations can be compared with each other [35, 40, 36, 34, 57]; however, a discussion of the discriminative power of the score functions is often neglected.

3.2 Calibration property

Uncalibrated raw 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 7 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 [35]. 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 FDR [35]. 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 [24, 38, 39] over various experimental protocols and diverse configurations of MS instruments. 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 employ analytical models, such as a binomial distribution in Andromeda [6] and MS

Amanda [10], Poisson distribution by Open Mass Spectrometry Search Algorithm (OMSSA) [20], a Weibull distribution for the XCorr [41], or a Gumbel distribution for Spectrum Specific P-value (SSPV) [56], 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 [8] 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.

X!Tandem [17] and Comet [14] 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 [17]. 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 [56].

Other types of p-value estimation methods exploit the exact null distribution obtained from scoring all possible peptide sequences that have the same − up to an instrument specific tolerance − precursor mass as the observed spectrum [38, 39, 40, 30, 45]. The explicit enumeration of all sequences is computationally unfeasible; therefore, dynamic programming technique is employed to count the peptides at each score in the null distribution. These methods, indeed, result in a perfect score calibration; however, they have several drawbacks.

1. They require proper estimation of the amino acid frequencies in the peptide database.

2. The calculation of the elements of the dynamic programming table requires a significant amount of CPU time.

3. The dynamic programming approach requires the score function to be additive [30].

4. 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. Note that, for low-resolution MS2 data, the information loss due to discretization hardly poses any problems in practice. This is discussed in details by Lin et al. [45].

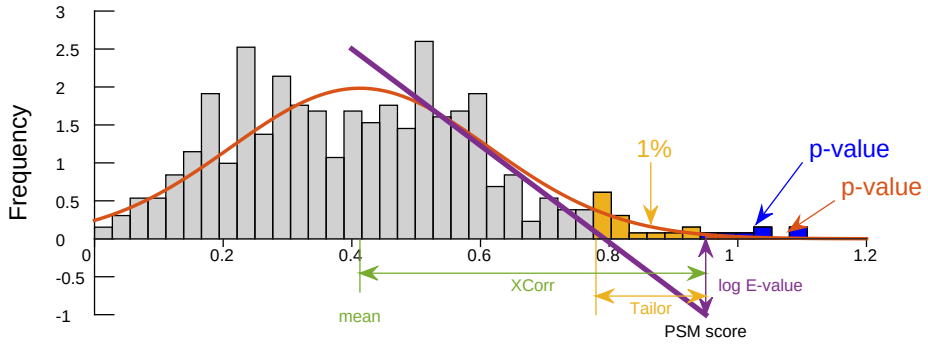

Figure 8: Illustration of the principles of the PSM score calibration approaches on a null distribution denoted by grey. The null distribution was obtained during scoring a real spectrum. (Green) XCorr calibrates the matching score by measuring the difference between the PSM score and an approximation 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 [35]. (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.

To overcome many of the issues mentioned above, empirical p-values of PSMs can be estimated via scoring spectra against a large number, say 10K, of decoy peptide databases [35]. 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 the expense of CPU time. Figure 8 illustrates and compares the principles of the score calibration methods on a null distribution.

3.3 Unbiasedness property

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.

3.4 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 instru-

ments and experimental protocols. For instance, features learnt 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.

3.5 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. 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.

4 Summary of the thesis articles

The main result of this thesis is the BoltzMatch method which is a stochastic neural network based scoring function for spectrum annotation. BoltzMatch was invented to provide a score function of high discriminative power. BoltzMatch models the joint probability of observing an experimental s and a theoretical h spectra, modelled with a restricted Boltzmann machine (RBM), and it is defined as

$$p(s,h) = \frac{1}{Z}\exp\{E(s,h)\}, \tag{9}$$

where the theoretical spectrum h is treated as an unobservable latent variable, an idealized version of the observed, flawed experimental spectrum s, which contains unexplainable peaks and incomplete fragmentation ion series. $E(s,h) = s^T W h$ is referred to as an energy function, and Z is a normalization factor which is defined as $Z = \sum_{s',h'} \exp\{E(s',h')\}$ for all possible vectors s', h', and 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.8. 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. Figure 9B illustrates the matching of an experimental and a theoretical spectra by XCorr and BoltzMatch.

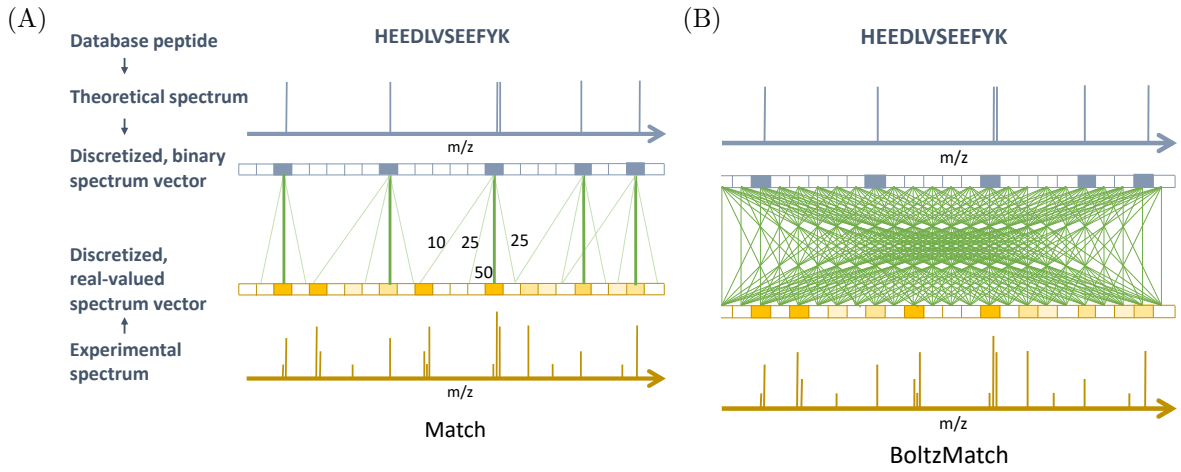

Figure 9: Graphical models of XCorr and BoltzMatch score functions. (A) XCorr weights matching ions by 50, flanking peaks by 25, and losses by 10; the weight values were specified manually. (B) Fully connected stochastic neural network, BoltzMatch, for matching observed with theoretical spectra. BoltzMatch considers the association between all peak pairs and learns to weight them solely from the data.

4.1 Training of BoltzMatch

Training of RBMs is carried out with maximum likelihood estimation

$$\tilde{w} = \underset{w}{\operatorname{argmax}} \log p_w(s) \tag{10}$$

where s denotes an experimental spectrum and p_w denotes a Gibbs distribution parametrized with w and modelled with a restricted Boltzmann machine. The weights are updated by calculating the derivatives of $\log p_w(s)$ with respect to the model parameters, that is,

$$w_{i,j}^{(t+1)} = w_{i,j}^{(t)} + \frac{\partial \log p_w(s)}{\partial w_{i,j}}, \tag{11}$$

where t indicates the iteration. The derivatives lead to

$$\frac{\partial \log p_w(s)}{\partial w_{i,j}} = \sum_{h'} p_w(h' \mid s)\ s[i]\ h'[j]\ - \sum_{s',h'} p_w(s',h')\ s'[j]\ h'[i], \tag{12}$$

where the first summation goes over all possible binary vectors h' and the second summation goes over all possible vectors of s' and h'. The conditional probability $p_w(h \mid s)$ is defined as $p_w(h = 1 \mid s) = \prod_i \sigma(\sum_{j=1}^{n} w_{ij} v_j)$, where $\sigma(a) = (1+\exp(-a))^{-1}$ is the sigmoid function [18]. The training of RBMs is notoriously hard (a) when latent variables are involved and (b) because it employs Markov chain Monte Carlo (MCMC) sampling to approximate the normalization factor Z [27] to avoid the intractable enumeration of s' and h'. In order to make the training of BoltzMatch more efficient, we developed a few tricks to tackle these problems by exploiting peculiarities of the mass spectrometry data:

1. Our model is restricted to only observed spectra s' and to possible theoretical spectra h' that encode real peptides. Moreover, we define $p_w(h', s') = 0$ whenever the precursor masses of these spectra are not equal up to an instrument-specific tolerance. Note that $p_w(h', s') = 0$ could lead to troubles when its logarithm is taken, thus, we just simply avoid considering such spectrum pairs in practice. Note that we consider this assumption reasonable in database-searching-base spectrum identification.

2. For every experimental spectrum there is only one theoretical peptide h considered that can be responsible for generating the observed spectrum s; therefore, we expect $p_w(s, h) >> 0$, while we expect $p_w(s, h') \approx 0$ for all other theoretical peptides h' within the precursor mass tolerance window. This will lead to a simplification of Eq.12 of the following form:

$$\frac{\partial \log p_w(s)}{\partial w_{i,j}} \approx \frac{\partial \log p_w(s,h)}{\partial w_{i,j}} = p_w(h \mid s)\ s[i]\ h[j]\ - \sum_{s',h'} p_w(s',h')\ s'[j]\ h'[i], \tag{13}$$

   where h is the theoretical peptide responsible for generating the observed spectrum s. Unfortunately, the correct theoretical peptide is not known. Therefore, a standard database searching step is carried out to identify the (possibly) correct theoretical spectrum for each experimental spectrum with a q-value less than 0.005 prior to the training of BoltzMatch.

3. The second summation of Eq.12 involves the enumeration of all possible vectors h'; however, most of them do not correspond to biologically plausible vector representations of any peptides. For instance, consider a vector h in which every second bin is filled with one while all

other bins are filled with zeroes; such vectors can be excluded from the enumeration. Therefore, we restrict the second summation to the candidate peptides of observed spectrum s, which leads to the following formula:

$$\frac{\partial \log p_w(s,h)}{\partial w_{i,j}} \approx p_w(h \mid s)\ s[i]\ h[j]\ -\ \sum_{h' \in CP(s)} p_w(s,h')\ s[j]\ h'[i], \tag{14}$$

where $CP(s)$ indicates the candidate peptides of the experimental spectrum s. Note that in general training of RBMs, the second summation is approximated using MCMC methods; however, in our opinion, the sampling would hardly result in any biologically plausible vector that could be associated with any real peptide molecule in this case.

4. Observed spectrum data set can contain ubiquitous peaks which appear in almost every spectrum at the same m/z location. For instance, the samples in the HumVar data set were prepared using TMT sixplex labelling which has an associated weight of 229.16293 Da and 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 labelling residues. These ubiquitous peaks do not contain useful information for spectrum identification, but they interfere in generative modelling as they can correlate with all other peaks. To mitigate the effect of these ubiquitous peaks, we added a diversifying regularization [58] − discussed in the next subsection in details − in the following form:

$$DR = \sum_{s_i, s_j \in MB} h_i^T h_j, \tag{15}$$

to the learning objective defined in Eq.14, 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 regularized training of BoltzMatch was carried out by optimizing $\log p_w(s,h)$ via maximum likelihood estimation:

$$\tilde{w} = \text{argmax}_w \left\{ \sum_{(s,h) \in D} \log \left(\frac{\exp(E_w(s,h))}{Z_s} \right) + \alpha DR \right\}, \tag{16}$$

where $Z_s = \sum_{h \in CP(s)} \exp E_w(s,h)$ and $(s,h) \in D$ denotes PSMs having q-values less than 0.005, which were obtained with standard database searching and α denotes a trade-off parameter of the regularization.

4.2 Diversifying regularization

The diversifying regularization (DR) was introduced as a general regularization method to help train arbitrary deep generative and discriminative models. The DR method was published as a separate article [58]. Here, I give brief summary of the method and the results.

Deep models [3], especially deep neural networks (DNNs), iteratively process data through various abstraction levels as $\mathbf{s} = \mathbf{h}_0 \rightarrow \mathbf{h}_1 \rightarrow \mathbf{h}_2 \rightarrow \cdots \rightarrow \mathbf{h}_L = y$. The first layer $\mathbf{s}$ is the raw data layer, and higher layers $\mathbf{h}_l$ aim to give a higher abstraction of the data, often referred to as features. The last layer can correspond either to class labels y in classification tasks or some other high-level cause in generative tasks. Each layer utilizes a monotone, non-linear so-called activation function $g_l(\mathbf{h}_l^T \theta_l) \rightarrow \mathbf{h}_{l+1}$ to transform features $\mathbf{h}_l$ to $\mathbf{h}_{l+1}$, where θ_l denotes the parametrization of the feature transformation at the given layer.

For a given set of data $\mathcal{D} = \{\mathbf{s}_i\}$, the model parameters are estimated by maximizing the data log likelihood, which is given as:

$$l(\theta; \mathcal{D}) = \log P(\mathcal{D}; \theta) = \sum_{\mathbf{s}_i \in \mathcal{D}} \log \sum_{\mathbf{h}} P(\mathbf{s}_i, \mathbf{h}; \theta). \tag{17}$$

Lower bound of Eq.17 could be written as following:

$$\mathcal{L}(\theta, \phi; \mathcal{D}) = \sum_{\mathbf{s} \in \mathcal{D}} \log P(\mathbf{s}; \theta) - \sum_{\mathbf{s} \in \mathcal{D}} KL\left(Q_{\mathbf{s}}^{\phi},\ P_{\mathbf{s}}^{\theta}\right). \tag{18}$$

This means that a tight lower bound can be achieved by minimizing the Kullback-Leibler (KL) divergence between the variational distribution Q and the exact posterior distribution P.

The training of θ parameters in deep models is notoriously hard, and it is often viewed as an art rather than a science. There are four main problems with training deep models for classification tasks:

I. Training of deep generative models via an unsupervised layer-wise manner does not utilize class labels, therefore essential information might be neglected.

II. When a generative model is learned, it is difficult to track the training, especially at higher levels [22]. For DNNs, the backpropagation method suffers from a problem known as vanishing gradients [16].

III. In principle, a generative model can be fitted to data arbitrarily well [59, 29, 26], in practice, the optimization procedure with latent variables can stuck in a poor local minima.

IV. The structure of the model is often specified in advance, and the designed model might not fit the data well. In particular, the number of hidden units or layers is often defined by the experimenter's intuition or habits; however, it is hard to give a bone fide estimation on the numbers of the latent components.

As a solution of aforementioned problems, a new regularization method was introduced, called diversifying regularization (DR), on the hidden units for training deep models for classification tasks. In principle, the proposed regularizer favours different abstract representation for two data samples belonging to different classes. This regularization is denoted by $D(\mathbf{h}_p^{(l)}, \mathbf{h}_q^{(l)})$, where $\mathbf{h}_p^{(l)}$ and $\mathbf{h}_q^{(l)}$ are abstract representations of data $\mathbf{s}_p$ and $\mathbf{s}_q$ (resp.) at layer l, and the data are of different types ($y_p \neq y_q$). For maximum likelihood estimation of generative models, DR is defined

in terms of divergence function $D(Q_{s_p}, Q_{s_q})$ and include it in Eq.18 as an additive term. For discriminative learning of DNNs using backpropagation, DR is defined as a distance function and include it in the learning objective as an additive cost.

As example of practical application, in deep belief networks (DBN) layers could be pre-trained with Restricted Boltzmann machines (RBMs).

The diversifying regularization for the variational optimization is introduced on the individual factors Q_s^j in the following way:

$$D_H(Q^{\phi}_{\mathbf{s}_p}, Q^{\phi}_{\mathbf{s}_q}) = 1 - \sum_{h=\{0,1\}} \sqrt{Q^{j,\phi}_{\mathbf{s}_p}(h) Q^{j,\phi}_{\mathbf{s}_q}(h)} \tag{19}$$

We tested the impact of DR on the training of a deep belief network (DBNs) [43] on the Cifar-10 dataset [42]. The dataset contains tiny (32×32) color pictures from 10 classes, 50,000 for training and 10,000 for testing. We constructed a DBN of 11 layers, each having 500, 300, 200, 150, 100, 80, 60, 50, 30, 20 hidden and 10 output units, respectively. The weights in every layer were pre-trained by RBM with DR and without DR as a baseline (control).

The results are shown in Figure 10. The first 10 plots (first five rows) show the pseudo-log likelihoods obtained during training RBM with DR (solid) and without DR (dashed) at every layer. The last two plots in the last row show the cost and the test error during the fine-tuning, when it was applied after regularized (solid) and unregularized (dashed) pre-training. These plots show that the fine-tuning achieved much faster convergence and much better generalization performance when the weights were pre-trained using DR. These results show that DR help the learner achieve better local minimum and lower generalization error.

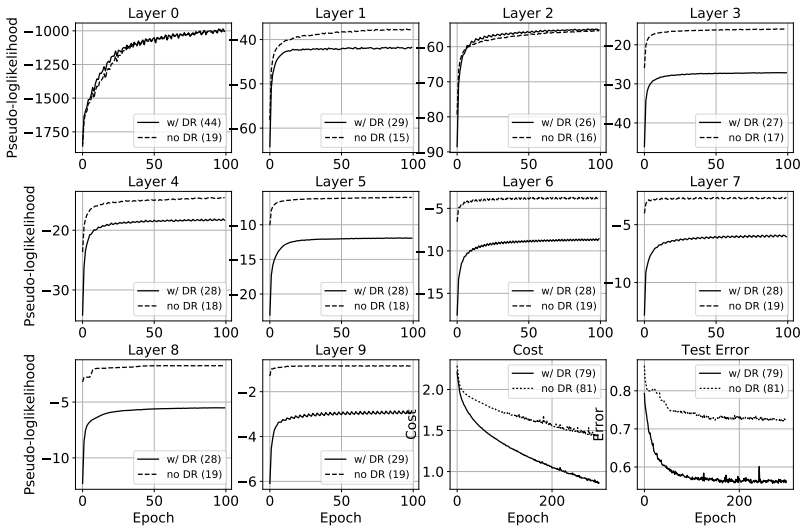

Figure 10: Learning curves during training of RBM and DBN. The numbers in the parenthesis in the legends indicate the run time in minutes. Cost is defined as cross entropy.

Overall, help of DR in case of DBNs could be explained since DR includes class information about the data, which might not maximize the pure log likelihood but favours solutions where different types of data have more different abstract representations as well. On the other hand,

discriminative training of deep models can also benefit from DR, because it can provide good gradients at low layers, directly helping to cope with vanishing gradient problem.

4.3 Evaluation of BoltzMatch in spectrum identification

Having BoltzMatch trained, PSMs were scored by $\log p(s,h) = E(s,h) - log(Z_s)$, where $Z_s = \sum_{h' \in CP(s)} E(s,h')$ and an experimental spectrum is annotated by the theoretical peptide $\tilde{h}$ that yields the highest score $\tilde{h} = \text{argmax}_{h' \in CP(s)} \log p(s,h')$. These scores are uncalibrated and proper score calibration methods can result in an increased number of spectrum annotations [35, 57]. To calibrate the BoltzMatch score with the XPV method, let $PV_s(c)$ denote the p-value of a score c for a given spectrum s calculated with the XPV method. Then the p-value of a spectrum s and a score $c = p(s,h)$ obtained with BoltzMatch can be derived as

$$\begin{aligned} PV_s(c) &= PV_s(p(s,h)) && (20)\\ &= PV_s(\log p(s,h)) && (21)\\ &= PV_s(E(s,h) - \log Z_s) && (22)\\ &= PV_s(E(s,h)) = PV_s((s^T W)h) && (23)\\ &= PV_s(s_{BM} h), && (24) \end{aligned}$$

where Eq.21 follows from the fact that the log function performs a monotone transformation, which does not have an impact on the p-value of any distributions, Eq.22 follows from the fact that the normalization factor $\log Z_s$ is a spectrum-dependent constant and can be omitted, $s_{BM} = s^T W$, and $s_{BM} h$ denotes the dot product of two vectors s_{BM} and h. Therefore, the BoltzMatch scores can be calibrated with any standard XPV score calibration methods using the transformed experimental spectra s_{BM}.

However, the XPV methods break down with high resolution MS2 data, which was discussed in Section 3.2 in point 4. We developed a new score calibration method, called Tailor, which works well not only with BoltzMatch, but with any score functions with high- and low-resolution information and Tailor does not rely on any assumptions on the form of the score distribution, i.e. whether it is e.g. binomial. This method will be introduced in details in the next section.

4.4 PSM score calibration with Tailor methods

The Tailor approach is a non-parametric, heuristic PSM score calibration method, which calibrates PSM scores by dividing them with the top 100-quantile of the empirical, spectrum-specific null distributions (i.e. the score with an associated p-value of 0.01 at the tail, hence the name) observed during database searching. Let us consider an experimental spectrum e 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 e is to be annotated with the peptide sequence that produces the score s_1. These scores form the basis of

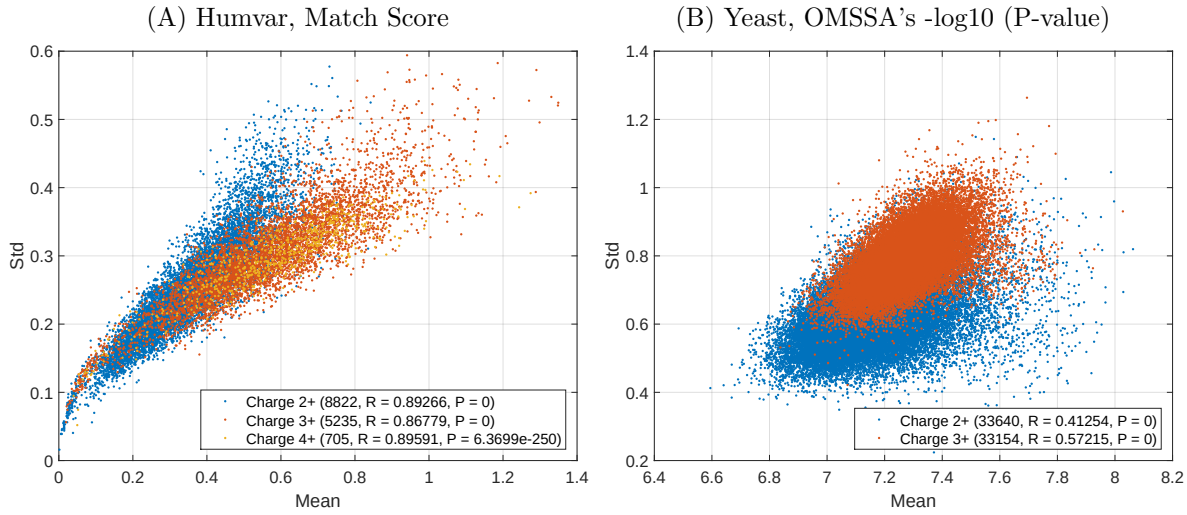

Figure 11: Correlation between the mean and the standard deviation (std) of the empirical null distributions. Each dot represents the mean and the std of the match scores of the candidate peptides of a single experimental spectrum. The scoring was carried out using the HumVar dataset with Match score (A) and Yeast with OMSSA's -log(p-values) (B) against a decoy peptide dataset. Colors indicate charge states; the numbers in parentheses show the number of the spectra, the correlation coefficients (R), and the p-values (P) for testing the hypothesis that there is no relationship between the mean and the std (null hypothesis).

an empirical null distribution for the spectrum e. The 100-quantiles define 99 cut points dividing the range of the probability distribution into 100, continuous intervals with equal probabilities. The last (99th) 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 match scores by

$$\tilde{s}_i = \frac{s_i}{Q100} \tag{25}$$

for $i = 1, \ldots, N$, which are simply referred to as Tailor scores.

Tailor method exploits the tail of the observed null distribution which may be inaccurate but random scores are observed during the database search step, but not the extreme tail, where samples are rare. This contrasts with exact p-value methods (XPV, MSGF+) which enumerate all random scores including at the extreme tail at the expense of CPU time to obtain an exact and accurate empirical null distribution. Therefore, Tailor is fast albeit less accurate, while exact methods are accurate, albeit slow.

The Tailor score calibration is based on division instead of subtraction, and this relies on the following empirical observation. The mean of the null distribution is highly correlated with its standard deviation, as illustrated in Figure 11 for two experimental datasets from our benchmark. Consequently, a null distribution that has a larger mean also has a tail that decays slower compared to distributions that have a smaller mean. Therefore, a certain difference, say l, between the top score s_1 and $Q100$ ($l = s_1 - Q100$) might be significant for null distributions with a low mean but can be puny for those having a large mean. The ratio of s_1 to $Q100$ implicitly takes into account

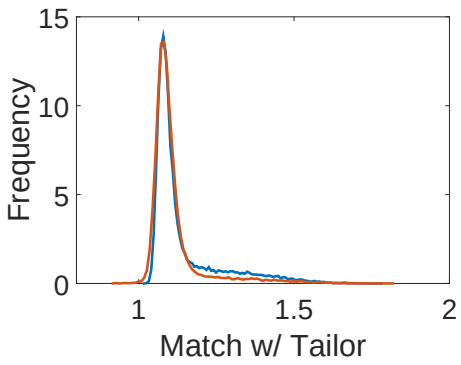

Figure 12: Distributions of top scoring PSMs obtained with match scores calibrated with the Tailor method on Yeast data for doubly (blue) and triply (red) charged precursor ions.

the width of the null distribution; that is, the wider the null distribution, the higher its mean, the higher the score Q100; therefore, the score s_1 is calibrated with a stronger factor $Q100$. Thus, the Tailor method incorporates the width (std) of the null distribution in this way. This contrasts with the XCorr metric.

An illustration on how the Tailor calibration decreases the difference between the score distributions of doubly and triply charged spectra is shown at Figure 12.

4.4.1 The main results of BoltzMatch in spectrum annotation

BoltzMatch was trained using three data sets obtained from previous publications, which contained high-resolution MS2 information (HumVar, iPRG, Malaria). They contained a total of 41,792 spectra and they were discretized with a 0.05 Da bin width. We note that BoltzMatch was trained and evaluated on these data sets separately, and the BoltzMatch search scores were calibrated with the Tailor method. Then we benchmarked BoltzMatch against several popular search engines and reported the number of accepted PSMs as a function of the q-values in Figure 6. The results show that BoltzMatch was able to annotate 12,019 observed spectra at a 0.1% FDR, which is 49.05% more annotations compared with the standard XCorr score function when both scorings were calibrated with the Tailor method. Conversely, XCorr annotated 12,019 spectra containing 5.3 times more false PSMs than BoltzMatch. Res-Ev method with XPV calibration is the current state-of-the-art scoring scheme designed specifically for high-resolution MS2 data, which was outperformed by BoltzMatch by around 17.78% more annotations at a 0.1% FDR; in contrast, Res-Ev yielded 12,019 PSMs with 4.4 times more errors than BoltzMatch.

4.5 Interpretation of BoltzMatch

To reveal the reasons why BoltzMatch outperforms XCorr, the transformed spectrum $s_{BM} = s^T W$ obtained with the weights of BoltzMatch was compared to s_{XC} obtained with the application of the cross-correlation penalty of XCorr [12] using an observed spectrum s from the Malaria data set (scan id = 7990). The two spectra are shown in Figures 13A-B.

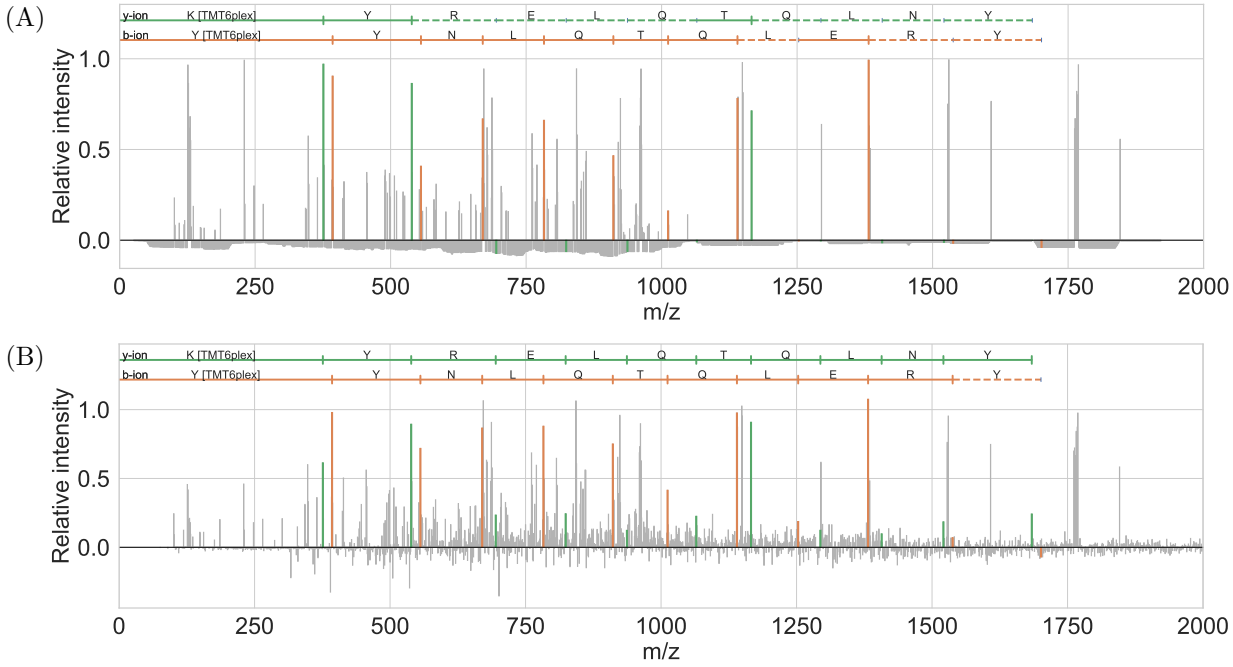

Figure 13: An observed spectrum from the Malaria data set (scan id = 7990) annotated with the database peptide YYNLQTQLERY. Peaks with positive intensity values matching to theoretical b- and y-ions are marked with green and orange colors, resp. (A) Annotation of s_{XC} obtained with XCORR with a q-value of 0.0049. (B) Annotation of s_{BM} obtained with BoltzMatch with a q-value of 0.0019.

On the one hand, this figure suggests that BoltzMatch normalizes the peak intensity depending on whether it can be explained by other nearby peaks, whereas XCorr diminishes peak intensities depending on the density of nearby peaks regardless of the semantic of their context. For instance, the peak corresponding to the b-ion YY (red peak near 600 m/z in Figure 13) was increased by around 55% with BoltzMatch but reduced by 10% with XCorr. On the other hand, BoltzMatch is able to recover peaks corresponding to unobserved but expected fragmentation ions. For instance, the peak corresponding to the y-ion KYR (green peak near 700 m/z in Figure 13) was recovered from its neighbouring peaks in s_{BM} with a positive intensity value, but it receives a negative intensity value (i.e. a penalty) in s_{XC} by XCorr.

4.6 Biased score functions

Recently, we showed that a simple machine-learning-based scoring system can easily learn to give preference to target peptides which in turn leads to a biased FDR estimation [8] as it was discussed in Section 3.3.

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 is slightly different in the spectrum vector space, and even 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 spectrum vectors, trained and tested on semi-tryptic peptides from Yeast protein sequences in which decoy peptides were produced by reversing. This means that scoring functions which consider account peak location specific weights can induce bias whether the weights are tuned manually or learned by a particular machine learning algorithm. The situation with vanilla artificial neural networks (ANNs) is even more dismal (or astonishing). On 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 14). 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 (resp. decoy) peptides are split randomly into positive and negative sets, the ANN achieves only 0.543 (resp. 0.512) AUC score (see Figures 3−5 in Supplementary to article "Bias in False Discovery Rate Estimation in Mass-Spectrometry-Based Peptide Identification"). 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 [25] can give preference to target peptides. The underlying problem is conceptual in our opinion. DRIP uses a set of spectra labelled by correct target peptides as training set to learn the parameters of a dynamic Bayesian network to model correct spectrum-peptide alignments. However, the training procedure also learns a bit of the distribution of the target peptides, resulting in a preference for them. We showed this by the following experiment using fully tryptic Yeast peptides. First, we generated 1000 spectrum-peptide pairs in silico to train DRIP. Then, we generated an additional 2000 synthetic spectra for searching with DRIP against a disjunct set of target and reversed decoy peptides. No correct PSMs were possible to find; thus, target and decoy peptides were expected to be equally likely to be assigned to query spectra. The XCorr scoring function achieved a fairly random assignment resulted in 0.496 AUC score. However, an ROC analysis of the search results of DRIP showed 0.531 AUC, which indicates target peptide preference (see orange and purple ROC lines in Figure 14).

4.6.1 Bias test of BoltzMatch

Here, I argue on theoretical and practical grounds that BoltzMatch is not biased.

For the theoretical ground, consider an observed spectrum s. BoltzMatch would be biased if it assigned roughly higher scores to target peptides than to decoy peptides, that is, $\log p(s,t) \gtrapprox \log p(s,d)$ for independently sampled target t and decoy d peptides, which are unrelated to s. This would imply that the Gibbs distribution represented by a restricted Boltzmann machine has higher mass around target peptides than around decoy peptides. However, target and decoy

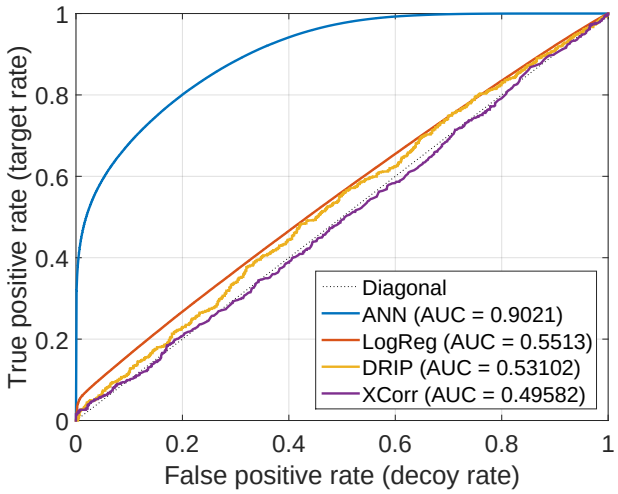

Figure 14: Discrimination of target against decoy peptides with artificial neural network (ANN, blue), logistic regression (LogReg, red), DRIP (orange), and XCorr scoring function (purple) evaluated by ROC analysis. The diagonal line (dashed line) indicates an perfectly unbiased scoring function, which would be the ideal case.

peptides are taken from the peptide data set $t, d \in CP(S)$, and they are treated equally in the - $\sum_{h' \in CP(s)} p_w(s, h')\ s[j]\ h'[i]$ phase (called negative phase). This means that the training procedure pushes down the unnormalized probability at t and d equally, if they are unrelated to s.

For the practical ground, all the 15,057 top-scoring PSMs were taken from the HumVar data set, which were obtained with BoltzMatch scoring, and selected the 5,000 worst-scoring PSMs (i.e., the bottom one-third from the ranked PSM list) for further ROC analysis. The tail of a ranked PSM list should contain only incorrect spectrum annotations, which are equally likely to be matched to either target or decoy peptides in case of an unbiased scoring method. Consequently, the distribution of the target PSM scores (i.e., PSMs in which spectra matched to target peptides) and the distribution of the decoy PSM scores should be indistinguishable, which can be tested with a ROC analysis. The area under the ROC curve (AUC) obtained with data used and shown in Figure 15 is 0.51 that has an associated p-value of 0.136 obtained with a two-sided Mann-Whitney U-test. This shows that one PSM score distribution is stochastically not greater than another one at a significance level $\alpha = 0.1$. The 2.000 worst-scoring PSMs of the ranked PSM list results in an AUC of 0.49927 with an associated p-value of 0.95522.

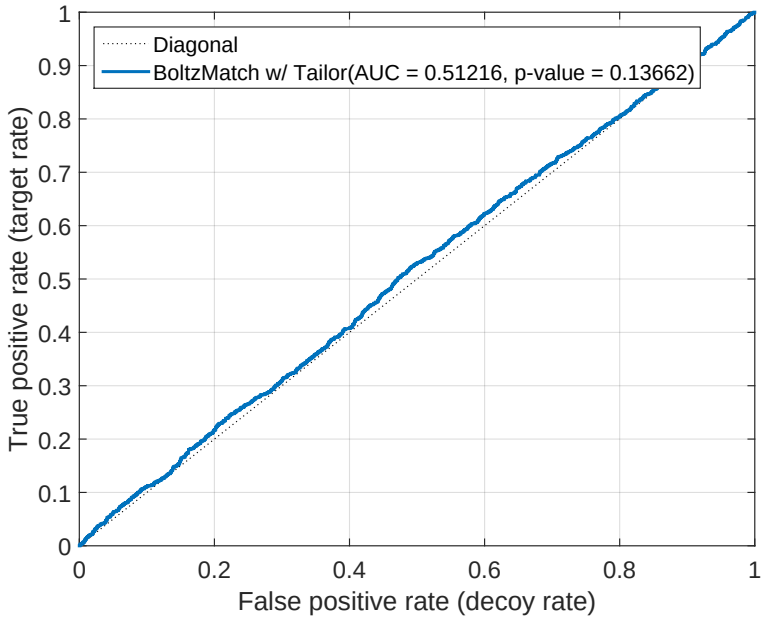

Figure 15: ROC analysis of the tail of a ranked PSM list obtained with scoring the HumVar dataset using a trained BoltzMatch. The diagonal line (dashed line) indicates an unbiased scoring function and identical distributions of the target and decoy PSM scores. The blue line shows the ROC analysis of the distribution of the target PSM scores against the distribution of the decoy PSM score. The p-value of ROC analysis was obtained with a two-sided Mann-Whitney U-test.

5 Conclusions

Many, if not all, machine learning-based methods for spectrum annotation uses self-supervised learning. In this approach, one performs a standard database searching to get some spectrum-peptide annotations as training data for a machine learning method, which in turn can be used to obtain, hopefully, more spectrum annotations. This self-supervised learning approach reminds us the chicken-egg problem, one needs annotated spectra to annotate more spectra. On one hand, this allows one to construct a set of training data immediately, for any instrument and experiment protocols using a fast and a simple scoring function, and the question remains whether the machine learning method is able to generalize from the training examples to annotate more spectra at any FDR level.

However, due to lack of human annotation, the training data lacks "hard examples" meaning spectrum-peptide matches which are truly correct but not straightforward to explain and which would be likely missed by standard score functions. Therefore, the main question remains whether BoltzMatch could be trained without labelled training data, i.e., could we train it with fully unsupervised fashion? Because BoltzMatch originates from restricted Boltzmann machines which can be trained via unsupervised fashion, BoltzMatch could be trained without annotated spectrum data as well.

I did try to train BoltzMatch without training data with the following modifications. In the modelling of the correct matches $p(s, h)$ − where the h were identified by standard database searching methods as defined in Eq.13 − the h was replaced with the filtered experimental spectrum s'; that is, the s' was obtained from s via removing the low intensity, possibly noisy peaks. The approach did result in an improved performance compared to the baseline, meaning that it managed to generalize to the baseline scoring methods XCorr. Unfortunately, this approach did not result in additional annotations compared to the case when self-supervision was used however, as for future research directions, it would be desirable to eliminate the self-supervision, and the target and decoy peptides from the learning pipelines because, this could reduce the risk of developing biased methods.

Bibliography

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

[2] P. Alves, R.J. Arnold, M.V. Novotny, P. Radivojac, J.P. Reilly, and H. Tang. Advancement in protein inference from shotgun proteomics using peptide detectability. *Pacific Symposium On Biocomputing*, pages 409–420, 2007.

[3] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[4] K. Biemann, C. Cone, B.R. Webster, and G.P. Arsenault. Determination of the amino acid sequence in oligopeptides by computer interpretation of their high-resolution mass spectra. *Journal of the American Chemical Society*, 88(23):5598–5606, 1966.

[5] Wikimedia Commons. Schematic representation of a tandem mass spectrometry experiment., 2006. Own work of K. Murray.

[6] 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.

[7] Alan Crooks. Mass spectrometer, 2010. SlideShare.

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

[9] Sven Degroeve and Lennart Martens. Ms2pip: a tool for ms/ms peak intensity prediction. *Bioinformatics*, 29(24):3199–3203, 2013.

[10] 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.

[11] Joshua E. Elias and Steven P Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods*, 3(4):207–214, 2007.

[12] Jimmy K Eng, Bernd Fischer, Jonas Grossmann, and Michael J MacCoss. A fast sequest cross correlation algorithm. *Journal of Proteome Research*, 7(10):4598–4602, 2008.

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

[14] 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.

[15] 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.

[16] Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal Vincent. The difficulty of training deep architectures and the effect of unsupervised pre-training. In *AISTATS*, volume 5, pages 153–160, 2009.

[17] 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.

[18] Asja Fischer and Christian Igel. An introduction to restricted boltzmann machines. In *Iberoamerican Congress on Pattern Recognition*, pages 14–36. Springer, 2012.

[19] Marjorie L. Fournier, Joshua M. Gilmore, Skylar A. Martin-Brown, and Michael P. Washburn. Multidimensional separations-based shotgun proteomics. *Chemical Reviews*, 107(8):3654–3686, 2007.

[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] Siegfried Gessulat, Tobias Schmidt, Daniel Paul Zolg, Patroklos Samaras, Karsten Schnatbaum, Johannes Zerweck, Tobias Knaute, Julia Rechenberger, Bernard Delanghe, Andreas Huhmer, Ulf Reimer, Hans-Christian Ehrlich, Stephan Aiche, Bernhard Kuster, and Mathias Wilhelm. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods*, 16(6):509–518, 2019.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

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

[24] 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.

[25] John T. Halloran, Jeff A. Bilmes, and William S. Noble. Learning peptide-spectrum alignment models for tandem mass spectrometry. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI14, pages 320–329, Arlington, Virginia, USA, 2014. AUAI Press.

[26] Eric J Hartman, James D Keeler, and Jacek M Kowalski. Layered neural networks with gaussian hidden units as universal approximations. *Neural comp.*, 2(2):210–215, 1990.

[27] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.

[28] CS Ho, MHM Chan, RCK Cheung, LK Law, LCW Lit, KF Ng, MWM Suen, and Tai HL. Electrospray ionisation mass spectrometry: Principles and clinical applications. *The Clinical biochemist. Reviews*, 24(1):3–12, 2003.

[29] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[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] K. Ishikawa and Y. Niwa. Computer-aided peptide sequencing by fast atom bombardment mass spectrometry. *Biological Mass Spectrometry*, 13(7):373–380, 1986.

[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] H.R. Kaufman, United States. National Aeronautics, Space Administration, and Lewis Research Center. *Performance Correlation for Electron-bombardment Ion Sources*. NASA technical note. National Aeronautics and Space Administration, 1965.

[34] Uri Keich, Attila Kertész-Farkas, and William Stafford Noble. Improved false discovery rate estimation procedure for shotgun proteomics. *Journal of Proteome Research*, 14(8):3148–3161, 2015.

[35] 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.

[36] Attila Kertész-Farkas, Uri Keich, and William Stafford Noble. Tandem mass spectrum identification via cascaded search. *Journal of proteome research*, 14(8):3027–3038, 2015.

[37] 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.

[38] 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.

[39] 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.

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

[41] 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.

[42] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

[43] Hugo Larochelle, Yoshua Bengio, Jérôme Louradour, and Pascal Lamblin. Exploring strategies for training deep neural networks. *JMLR*, 10(Jan):1–40, 2009.

[44] Lev I. Levitsky, Mark V. Ivanov, Anna A. Lobas, and Mikhail V. Gorshkov. Unbiased false discovery rate estimation for shotgun proteomics based on the target-decoy approach. *Journal of Proteome Research*, 16(2):393–397, 2017.

[45] 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.

[46] Bingwen Lu and Ting Chen. Algorithms for de novo peptide sequencing using tandem mass spectrometry. *Drug Discovery Today*, 2(2):85–90, 2004.

[47] Alexey I Nesvizhskii and Ruedi Aebersold. Analysis, statistical validation and dissemination of large-scale proteomics datasets generated by tandem ms. *Drug discovery today*, 9(4):173–181, 2004.

[48] Tran Ngoc Hieu, Zhang Xianglilan, Xin Lei, Shan Baozhen, and Li Ming. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, 114(31):8247–8252, 2017.

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

[50] 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.

[51] J.J. Pitt. Principles and applications of liquid chromatography-mass spectrometry in clinical biochemistry. *The Clinical Biochemist. Reviews*, 30(1):19–34, 2009.

[52] Donald Voet Pratt, G. Voet Judith, and W. Charlotte. *Fundamentals of biochemistry : life at the molecular level*. Hoboken, NJ: Wiley, 2006.

[53] Jesse Rodriguez, Nitin Gupta, Richard D. Smith, and Pavel A. Pevzner. Does trypsin cut before proline? *Journal of Proteome Research*, 7(1):300–305, 2008.

[54] B. Shin, H.J. Jung, S.W. Hyung, H. Kim, D. Lee, C. Lee, M.H. Yu, and S.W. Lee. Postexperiment monoisotopic mass filtering and refinement (pe-mmr) of tandem mass spectrometric data increases accuracy of peptide identification in lc/ms/ms. *Mol Cell Proteomics*, 7(6):1124–1134, 2008.

[55] H. B. Sieburg. Physiological studies in silico. *Studies in the Sciences of Complexity*, 12:321–342, 1990.

[56] 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.

[57] Pavel Sulimov and Attila Kertsz-Farkas. Tailor: A nonparametric and rapid score calibration method for database search-based peptide identification in shotgun proteomics. *Journal of Proteome Research*, 19(4):1481–1490, 2020.

[58] Pavel Sulimov, Elena Sukmanova, Roman Chereshnev, and Attila Kertesz-Farkas. *Guided Layer-Wise Learning for Deep Models Using Side Information*, pages 50–61. Springer, 02 2020.

[59] Ilya Sutskever and Geoffrey E. Hinton. Deep, narrow sigmoid belief networks are universal approximators. *Neural Computation*, 20:2629–2636, 2008.

[60] 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.

[61] M.R. Wilkins, R.D. Appel, K.L. Williams, and D.F. Hochstrasser. *Proteome Research Concepts, Technology and Application*. Springer, 2007.

[62] Dirk A. Wolters, Michael P. Washburn, and John R. Yates. An automated multidimensional protein identification technology for shotgun proteomics. *Analytical Chemistry*, 73(23):5683–5690, 2001.

[63] 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.

[64] Xie-Xuan Zhou, Wen-Feng Zeng, Hao Chi, Chunjie Luo, Chao Liu, Jianfeng Zhan, Si-Min He, and Zhifei Zhang. pdeep: Predicting ms/ms spectra of peptides with deep learning. *Analytical Chemistry*, 89(23):12690–12697, 2017.