Computational methods to estimate error rates for peptide identifications in mass spectrometry-based proteomics

XIAO LIANG

Master of Science Thesis
Stockholm, Sweden 2013
Computational methods to estimate error rates for peptide identifications in mass spectrometry-based proteomics

XIAO LIANG

Master’s Thesis in Numerical Analysis (30 ECTS credits)
Master Programme in Computer simulation for Science and Engineering (120 credits)
Royal Institute of Technology year 2013
Supervisor at KTH was Lukas Käll
Examiner was Michael Hanke

TRITA-MAT-E 2013:32
ISRN-KTH/MAT/E--13/32--SE
Abstract

In the field of proteomics, tandem mass spectrometry is the core technology which promises to identify peptide components within complex mixtures on a large scale. Currently the bottleneck is to reduce the error rates and assign accurate statistical estimates of peptide identifications.

In this work, we introduce the techniques of identifying chimeric spectra, where two or more precursor ions with similar mass and retention time are co-fragmented and sequenced by the MS/MS instrument. Based on this, we try to analyze the factor which leads to the high error rate of identifications. We show that chimeric spectra have high correlations with the ranking scores and can reduce the number of positive identifications.

Additionally, we address the problem of assigning a posterior error probability (PEP) to the individual peptide-spectrum matches (PSMs) that are obtained via search engines. This problem is computationally more difficult than estimating the error rate associated with a large collection of PSMs, such as false discovery rate (FDR). Existing methods rely on parametric or semi-parametric models of the underlying score distribution as a preassumption. We provide a so-called kernel logistic regression procedure without any explicit assumptions about the score distribution. Based on an appropriate positive definite Gaussian kernel, the resulting PEP estimate is proven to be robust by achieving a close correspondence between the PEP-derived q-values and FDR-derived q-values. Furthermore, we also accept at least 200 more significant PSMs with setting a threshold based on PEP-derived q-values compared to FDR-derived q-values. Finally, we show that this kernel logistic regression method is well established in the statistics literature and it can produce accurate PEP estimates for different types of PSM score functions and data.
Sammanfattning

Tandemmasspektrometri (MS/MS) är kärnan i proteomikstudier som försöker att identifiera peptider inom komplexa proteinlösningar i stor skala. För närvarande är flaskhalsen att minska felprocenten av peptideidentifikationerna, samt att tilldela noggranna statistiska skattningar av dessa.

I detta arbete presenterar vi metoder för att identifiera chimära spektra, där två eller flera produktjoner med liknande massa och retentionstid är samfragmenterade och sekvenserade i ett MS/MS-instrument. Hypotesen är att dessa samfragmenterade joner är en anledning till den höga felfrekvensen hos peptideidentifikationer. Vi visar att chimära spektra har korrelerar med identifikationskvaliteten och kan minska antalet positiva identifieringar.

# Contents

1 Introduction ................................................................. 1
  1.1 Background ......................................................... 1
  1.2 Peptide identification process in shotgun proteomics ............ 3
    1.2.1 Shotgun proteomics experiments .......................... 3
    1.2.2 Database searching methods .............................. 4
    1.2.3 Decoy Databases ........................................... 6
  1.3 Post-processors for improving identifications .................. 7
    1.3.1 Percolator - semisupervised machine learning method ...... 7
  1.4 Error rates for peptide to spectrum matches ...................... 10
    1.4.1 False discovery rate .................................... 11
    1.4.2 $q$-value .................................................. 11
    1.4.3 Posterior error probability ............................... 11

2 Methods ........................................................................... 13
  2.1 Data collection process ............................................. 13
  2.2 Chimeric MS/MS spectra ............................................ 13
  2.3 Non-parametric model to estimate PEP ............................ 16
    2.3.1 Modelling ...................................................... 16
    2.3.2 Roughness penalty approach ............................... 17
  2.4 Estimate PEP using kernel logistic regression .................... 19
    2.4.1 Mercer kernel ................................................ 19
    2.4.2 Kernel logistic regression model .......................... 20
    2.4.3 Solving linear equations .................................... 23
    2.4.4 Generalized cross-validation based model selection strategy .. 23
  2.5 Accuracy of the PEP estimates ....................................... 24

3 Results ............................................................................ 27
  3.1 Impact of chimeric spectra .......................................... 27
  3.2 PEP estimates .......................................................... 27
  3.3 Comparison of $q$-values ............................................ 28
  3.4 Generalized property of KLR method ............................. 29
  3.5 Software: Qvality-KLR ............................................... 30
Chapter 1

Introduction

1.1 Background

In large-scale proteomics analysis, shotgun proteomics involving tandem mass spectrometry is widely used to identify proteins in complex mixtures (Aebersold and Mann, 2003; Noble and MacCoss, 2012). Tandem mass spectrometry, also known as MS/MS or MS2, includes two steps of selections, with some form of fragmentation occurring in between the stages. The intact peptide masses are scanned in the first mass analyzer, and are allowed to fragment and then all resulting masses are scanned in the second analyzer (Nesvizhskii, 2010).

Owing to the mixture and fragmentation, the number of spectra data from the high-throughput proteomics experiments is so huge that it requires efficient and accurate methods of mapping from MS data to the corresponding peptides.

During the past decade, a number of powerful tools have been developed for solving the problem, such as Crux (Park et al., 2008), SEQUEST (Eng et al., 1994; López-Ferrer et al., 2004), Mascot (Perkins et al., 1999), which search a given sequence database and output a collection of correct or incorrect peptide-spectrum matches (PSMs) with associated scores. Usually a threshold on the scores is used to rule out the incorrect PSMs. However, the rate of incorrect peptide assignments can be very high. Such database searching methods are still widely used in this field due to the high efficiency although many spectra are incorrectly matched. Therefore, the core problem in this field becomes to reduce the high error rate of peptide identifications.

The results from current scoring methods show a significant overlap between the score distributions of correct and incorrect peptide identifications (Noble and MacCoss, 2012; Nesvizhskii, 2010; MacCoss et al., 2002; Keller et al., 2002). For this problem, an alternative machine learning method Percolator (Käll et al., 2007a) is also chosen as a post-processor to discriminate between the correct and incorrect PSMs by re-ranking the scores and training classifying datasets. This approach utilizes the support vector machine algorithm with use of several features based on the PSMs scores and show up a significant optimization.
In the last years a lot of research effort went into the assessment of the quality and reliability of peptide and protein identification results. For this a lot of statistical assessments are developed (Nesvizhskii, 2010; Käll et al., 2007b; Granholm and Käll, 2011). The idea behind this is to set an appropriate threshold based on different statistical assessments instead of database searching scores and hope to obtain more reliable identifications. Currently different statistical scores can be used to show the confidence of an identification and a set of methods are developed to estimate these scores (Granholm and Käll, 2011). It suggests that it is more meaningful to report false discovery rates (FDRs) or q values for sets of PSMs, and Posterior error probability (PEPs) for individual PSMs, rather than reporting metrics without multiple testing corrections.

In this work, we firstly introduce the techniques of identifying peptides from MS1 and MS2 data. We focus on the first-step spectra before the fragmentation and the impact of co-fragmentation on the identification results. From which we hope to find an appropriate way to improve the quality of identifications.

The purpose of this project is also and primarily to describe how well-established statistical methods for significance analysis can be applied to peptides identified via tandem mass spectrometry. Assuming that we set up an appropriate peptide identification method and output a list of PSMs ranked by some score. We then need to provide statistical measurements for the significance of the PSMs. As a general property, the FDR can be estimated simply by the average PEP of the significant PSMs. But conversely, the PEP is related to the derivative of the FDR which is usually inobvious. Although calculating PEP is not that easy, it is a convenient way to explain how good the identifications are and do further convergence analysis for different tools. Existing methods rely on parametric or semi-parametric models of the underlying score distribution as an assumption. For this problem, we provide a different way to estimate the PEP for each PSM and try to avoid any pre-assumptions of the score distributions.
1.2 Peptide identification process in shotgun proteomics

1.2.1 Shotgun proteomics experiments

Shotgun proteomic experiments generate high-throughput data in complex mixtures using a combination of liquid chromatography with mass spectrometry. Thus it is widely used for the large-scale peptide and protein identification (Aebersold and Mann, 2003). There are several preparation steps which need to be mentioned before we get the expected data for peptide identification (Nesvizhskii, 2010).

The first step is to digest sample proteins into peptides using enzymes such as trypsin. Resulting peptide mixtures are optionally processed to capture a particular class of peptides (e.g. phosphorylated peptides), and then separated using a liquid chromatography (LC) system coupled online to a mass spectrometer. Performing protein digestion has many advantages, while as a drawback, since each protein digested with trypsin produces multiple peptides, the resulting peptide mixtures can be very complex. Alternatively, prior to digestion, a protein separation procedure may be employed.

After the preparation, the next step is peptide sequencing using tandem mass spectrometry (MS/MS). At this stage, the digested peptides are ionized at a certain retention time and transferred into the gas phase. Ions are then selected and subjected to tandem mass spectrometry (MS/MS) sequencing to generate the final MS/MS spectra. This process particularly consists of the following two stages: 1) The instrument firstly scans all of the peptide ions which are introduced at any time and then records the so-called MS1 spectrum which consists of two-column data: mass-to-charge ratios and intensities of all peptide ions. 2) There is also an automated selection procedure by the instrument. Selected peptide ions (precursor or parent ions) are broken down into smaller pieces (fragment ions) in the collision cell of the MS instrument. Now the MS2 spectrum is acquired which also consists of a list of mass-to-charge (m/z) values and intensities of all the fragment ions generated by fragmenting an isolated precursor ion.

Why the fragmentation? This pattern actually allows identification of the amino acid sequence of the peptide that produced it. For the subsequent peptide identification step, the fragmented spectrum is easier to be analyzed by tools. Therefore, it is also more widely used compared to the MS1 spectrum. As described above, we could imagine that the mass accuracy of the analyzer in fact has a significant effect on the acquired spectra. Because an MS instrument can measure peptide ion m/z values ranges from as low as several parts per million (ppm), such as LTQ-Orbitrap, to more than 500 ppm in the case of low mass accuracy instruments.

Figure 1.1 gives an overview of the shotgun proteomics procedure. Due to the accuracy of instruments and the fragmentation procedure, a lot of noise could be detected and the intensities for certain masses could be completely different.
1.2.2 Database searching methods

After acquiring a huge amount of mass spectrometry data from a shotgun proteomics experiment, the problem now shifts to the identification of the peptides which give rise to the measured MS/MS spectra. Peptide identification can be solved by comparing MS/MS spectra with theoretical spectra of peptides calculated by a protein sequences database, or against spectra from an existing spectral library. Peptide sequences can also be extracted from the spectra using de novo sequencing.

A lot of work and effort has been done to construct and enrich the protein databases. Thus, the former database search approach is the most popular way to identify peptides at this stage. The idea behind this approach is to measure the similarity between the MS/MS spectrum and the calculated theoretical spectrum. The most widely used methods, such as SEQUEST (Eng et al., 1994), Mascot (Perkins et al., 1999), are those which search the databases for the peptide whose theoretical spectrum matches the observed spectrum best. The resulting peptide-spectrum matches (PSMs) are ranked by a scoring function, which shows a way to judge whether a result is significant or not. Thus, except choosing an appropriate database to do the matches, the subsequent problem is the choice of a good measurement or scoring function. Figure 1.2 presents a flowchart of this procedure.

### Scoring function

Each scoring function is based on a measurement of the similarity distance between the experimental MS2 spectrum and the theoretical spectrum. The score could be a cross-correlation, a probability, or a simple dot product, any arbitrary similarity measure.

As an example, cross-correlation score (XCorr) and Sp score in SEQUEST give the most widely used measurements of the matches. As an improvement and integration of variant tools in this field, Crux has been developed (Park et al., 2008). In this work, it implements SEQUEST in the beginning to retrieve all the candidate peptides for a given MS2 spectrum. The resulting spectrum is compared with the
1.2. PEPTIDE IDENTIFICATION PROCESS IN SHOTGUN PROTEOMICS

Peptide MS/MS

Observed spectrum

m/z

Theoretical spectrum

Compare

Protein sequence database

intensity

ranked peptide match lists

peptide        score
VSTTTVDL       6.1
CSLLAPL        1.0
LSQASAPL      0.5

Peptide to spectrum matches (PSMs)

Figure 1.2. Overview of database searching procedure.

fragment ions \( u \) from the candidate peptides. All the PSMs are ranked according to the score \( S_p \) as follows:

\[
S_p(u, v) = \frac{\sum_{i=1}^{N} v_i \delta(u_i)}{1 + 0.075 R(u, v)}
\]

where \( N \) is the maximal mass-to-charge, \( R(u, v) \) is the maximum number of consecutive b- or y-ions from the theoretical spectrum that appear in the observed spectrum, \( \delta(x) = 0 \) when \( x = 0 \), otherwise \( \delta(x) = 1 \).

Subsequently, the 500 top-scoring PSMs are again ranked according to score \( X_{corr} \):

\[
X_{corr}(u, v) = \frac{0.015 \sum_{i=1}^{N} u_i v_i}{\sum_{\tau=-75}^{75} \sum_{i=1}^{N} u_i v_{i-\tau}}
\]

After the database searching procedure, Crux generates an sqt-format output which includes all the search results for each spectrum (an example provided in Fig 1.3). It consists of H, S, M, and L lines. It has been proved that this format can reduce the disk space significantly (McDonald et al., 2004).
Figure 1.3. Sqt file format example header. The result for each spectrum starts with an S line which gives the scan number of this spectrum and other metrics. The M line contains the highest scores for the search results and is followed by one or more L lines which give the corresponding peptide information.

1.2.3 Decoy Databases

The idea of decoy database searching is really simple but powerful for estimating the statistical significance of PSMs (Moore et al., 2002). A decoy database is a database of amino acid sequences generated deliberately by reversing or shuffling the target sequences. It can also be generated at random by using a Markov model with certain parameters based on the target sequences. In this way, we can ensure the decoy database only contains peptide-like amino acid sequences which are not in the original target database. If we search spectrum against the decoy database, the resulting PSM must be incorrect, even if it has a high score.

Figure 1.4 shows the distributions of XCorr from a database-searching software Crux when searching against the target and decoy databases.

The distribution of target PSM scores shows a higher tail to the right. This explains the well-acted simulation of the null distribution by decoy searching strategy on the other side. It is clear that the overlap between the scores for correct and incorrect peptide identifications reduces the accuracy. This obstruction limits the identification to either eliminating a large number of true positive identifications to decrease the false positives or allowing a large number of false positive identifications to increase the number of true positive identifications.
1.3 Post-processors for improving identifications

Identifying peptides based on tandem mass spectrometry via a search engine is essentially a ranking procedure. We are more interested in the good PSMs which are ranked higher than other PSMs. Empirical evidence suggests that, although Xcorr does a good job at relative ranking, it fails when attempting to rank PSMs absolutely (Keller et al., 2002). Namely, we can not completely rely on Xcorr scores and accept all the PSMs which have a relative high Xcorr scores. This also happens to other types of search engine scores. Therefore methods for post-processing PSMs have been developed to improve their absolute ranking.

A semi-supervised machine learning method called Percolator is given in Käll et al. (2007a) to learn an absolute PSM ranking function dynamically with use of the target-decoy strategy.

1.3.1 Percolator - semisupervised machine learning method

Percolator is essentially a dynamic classification algorithm. Each PSM is assigned a collection of features which capture characteristics of the corresponding spectrum, the peptide, and the quality of the match between the spectrum and the peptide.

Figure 1.4. An example for distributions of XCorr for the target and decoy PSMs. The decoy line simulates the distribution of the lower scores for target PSMs. The target PSMs have more higher scores in the right tail.
With use of the information, Percolator attempts to maximize the number of peptides identified at a target false discovery rate by training a discriminative classification algorithm which follows the method of linear SVMs with $L_2$ loss function described in (Keerthi and DeCoste, 2006). Using this semi-supervised machine learning method, we hope to obtain a new score distribution with less overlapping region in between the correct and incorrect PSMs (Figure 1.5).

In particular, Percolator stores the top-scoring PSM against target and decoy database and computes a vector of features ($X_{corr}$, $Sp$, etc.) for each spectrum. It randomly divides the set of decoy PSMs in half: one for training, the remainder for unbiased estimate. The core phase in Percolator algorithm consists of three steps: 1) selecting a subset of positive PSMs with use of a given threshold of false discovery rate, 2) training an support vector machine to maximize the distance between positive and decoy PSMs 3) re-ranking the entire set of target and decoy PSMs using the training result. The algorithm is given in 1.

**Support vector machine $L_2$-SVM**

Consider a binary classification problem $\{x_i, t_i\}_{i=1}^N$, where $x_i$ is a score and $t_i \in \{-1, +1\}$. In our case, $x_i$ is a target PSM score when $t_i = +1$, otherwise $x_i$ is from the decoy database searching. To obtain a linear classifier $y = w \cdot x + b$, we solve the following minimization problem (Keerthi and DeCoste, 2006):

$$
\min_{\beta} f(\beta) = \min_{\beta} \left\{ \frac{\lambda}{2} \| \beta \|^2 + \frac{1}{2} \sum_{i \in I(\beta)} d_i^2(\beta) \right\},
$$

(1.3)

where $\beta = (w, b)$, $\lambda = 1/C$, $d_i(\beta) = y_i(\beta) - t_i$, $y_i(\beta) = w \cdot x_i + b$, and $I(\beta) = \{i : t_i y_i(\beta) < 1\}$.

First note that $f$ is a piecewise quadratic function and the presence of the first term $\lambda/2\| \beta \|^2$ makes $f$ strictly convex. Thus it has a unique minimizer. Second,
Algorithm 1 Percolator algorithm. Input: \( T \) is the output of search engine against a target database, \( D \) is the output against decoy database. If we use Crux, then it should be as Figure 1.3. \( t \) is set as the desired false discovery rate.

1: procedure PERCOLATOR((\( T, D, t, \text{iteration} \))
2: \( F_t \leftarrow \text{computeFeatures}(T) \) \( \triangleright \) Compute the feature vector.
3: \( F_d \leftarrow \text{computeFeatures}(D) \)
4: for \( j \leftarrow \{1, 2, \ldots, \text{iteration} \} \) do
5: \( F_t^+ \leftarrow \text{selectByFDR}(t, F_t, F_d, T, D) \) \( \triangleright \) Select the positive PSMs.
6: \( W \leftarrow \text{trainSVM}(F_t^+, F_d) \) \( \triangleright \) Train the classifier.
7: \( X_t \leftarrow \text{rerank}(W, F_t) \)
8: \( X_d \leftarrow \text{rerank}(W, F_d) \)
9: end for
10: return (\( \text{selectByFDR}(t, F_t, F_d, T, D) \))
11: end procedure

\( f \) is continuously differentiable in spite of the jumps in \( I(\beta) \), since \( d_i = 0 \) when it comes to a jump.

Now let us define the function \( f_I \) for an given index set \( I \subset \{1, \ldots, m \} \):
\[
f_I(\beta) = \frac{\lambda}{2} \|\beta\|^2 + \frac{1}{2} \sum_{i \in I} d_i^2(\beta). \tag{1.4}
\]

\( f_I \) and \( f \) shares the same properties as well as the gradient: \( \nabla f(\beta)|_{\beta=\hat{\beta}} = \nabla f_I(\beta)|_{\beta=\hat{\beta}} \) where \( \hat{I} = I(\hat{\beta}) \), since there exists an open set in which \( f \) and \( f_I \) are identical. It follows that \( \hat{\beta} \) minimizes \( f \) if and only if it minimizes \( f_I \).

Solving Linear system using CGLS algorithm

Now the problem comes to the minimization of \( f_I \). The solution can be approached in any of the following way: 1) using factorization methods such as QR or singular value decomposition (SVD); 2) using an iterative method such as conjugate gradient (CG) method. Due to the large-scale data with sparse structure, we prefer the latter approach. CG method is also able to make effective use of knowledge of good starting vectors. Let us consider the regularized least squares problem:
\[
f_I(\beta) = \frac{\lambda}{2} \|\beta\|^2 + \frac{1}{2} \|X\beta - t\|^2 \quad \text{where} \quad X_i = (x_i^T, 1), i \in I.
\]
Furthermore, this corresponds to a normal system:
\[
(\lambda I + X^T X)\beta = X^T t. \tag{1.5}
\]

For the work of \( L_2 \)-SVM used in Percolator, the CGLS algorithm is given by Frommer and Maass (1999). CGLS is a special conjugate-gradient solver that is designed to solve large-scale, sparse, weighted, regularized least-squares problems in a numerically robust way (see Algorithm 2).
Algorithm 2 CGLS algorithm. In the algorithm, $-z^j$ is the vector containing the classifier residuals $d_i$. $r^j$ is the negative of the gradient of $f_1(\beta)$ at $\beta^j$.

1: procedure CGLS($\beta^0$)
2: $z^0 \leftarrow t - X\beta^0$ \Comment{Initialize parameters}
3: $r^0 \leftarrow X^T z^0 - \lambda \beta^0$
4: $p^0 \leftarrow r^0$
5: for $j \leftarrow \{0, 1, 2, \ldots\}$ do
6: \hspace{1em} $q^j \leftarrow X p^j$
7: \hspace{1em} $\gamma^j \leftarrow ||r^j||^2 / (||q^j||^2 + \lambda ||p^j||^2)$
8: \hspace{1em} $\beta^{j+1} \leftarrow \beta^j + \gamma^j p^j$
9: \hspace{1em} $z^{j+1} \leftarrow z^j - \gamma^j q^j$
10: \hspace{1em} $r^{j+1} \leftarrow X^T z^{j+1} - \lambda \beta^{j+1}$
11: \hspace{1em} $\omega^j \leftarrow ||r^{j+1}||^2 / ||r^j||^2$
12: \hspace{1em} $p^{j+1} \leftarrow r^{j+1} + \omega^j p^j$
13: end for
14: $r^{j+1} = 0 \leftarrow$ check convergence
15: return ($\hat{\beta}$)
16: end procedure

1.4 Error rates for peptide to spectrum matches

Because of an increasing size of experimental datasets, proteomic research is increasingly dependent upon the automated computational tools described above. These tools generate a large amount of peptide to spectrum matches (PSMs) of variable quality and majority matches are incorrect. However, such a high error rate is mainly caused by the instruments of low accuracy and it is hard to be reduced by computational and analysis tools. Thus, an associated statistical confidence measurement of PSMs is required in order to interpret the computational results.

Recalling the procedure of database searching methods, a threshold score is always employed after sorting the PSMs according to their scores obtained from the search engine. A PSM scoring higher than the threshold is regarded as significant. Our goal is to convert these scores into a more useful set of significance measures. With these measures, we could simply say which PSM is more likely to be correct.

Although the process of assessing the validity of a PSM is not limited to the information contained in the database search score. The discussion below will start with a focus on the database search scores as the primary source to develop the methods, since it is the most straightforward way to present each PSM. In this section, we introduce three frequently used statistical scores: the false discovery rate (FDR), $q$-value and posterior error probability (PEP).
1.4. ERROR RATES FOR PEPTIDE TO SPECTRUM MATCHES

1.4.1 False discovery rate

FDR is defined as the expected ratio of false positives among the significant test statistics (Käll et al., 2007c; Storey and Tibshirani, 2003; Benjamini and Hochberg, 1995). For the case of shotgun proteomics, the FDR is specifically the expected fraction of incorrect PSMs among the accepted PSMs scoring above the threshold.

As a summarized statistics for the whole collection of PSMs, the FDR can be estimated by using a target-decoy strategy. The matches which are obtained through searching against a decoy database are assumed to be always incorrect. Then FDRs can be estimated with use of decoy PSMs. Denote $x^t$, $D$ and $T$ to be the score threshold, decoy and target database respectively, then the corresponding FDR is

$$FDR(x^t) = \pi_0 \frac{|\{y|y \geq x^t, y \in D\}|}{|\{y|y \geq x^t, y \in T\}|},$$

where $\pi_0$ represents the prior probability of all the incorrect matches (decoy PSMs in this case).

1.4.2 q-value

However, the estimate of FDRs is not guaranteed to increase monotonically with the PSM quality. Thus, in order to assign statistical scores to individual PSMs, the $q$ value is then defined as the minimal FDR value of any dataset which includes the current PSM. So the FDR-derived q-value has the following form:

$$q_{FDR}(x^t) = \min_{x \geq x^t} \pi_0 \frac{|\{y|y \geq x, y \in D\}|}{|\{y|y \geq x, y \in T\}|}$$

A dataset with a $q$ value equal 0.01 is empirically considered as a description of all PSMs with the same score or higher. Hence we could pick an exact point as the accepted threshold after sorting by $q$ value. The lower $q$ value we choose, the less number of incorrect PSMs above the threshold we could expect.

1.4.3 Posterior error probability

From the probabilistic perspective, the posterior error probability represents how likely a single PSM is actually incorrect. With analysis of the distribution of the whole datasets in more details, we could sometimes obtain a completely different PEP value of a certain PSM when compared to the FDR value of its corresponding set.

The PEP is known as local FDR (or fdr for short) (Efron et al., 2001), while it is usually much more difficult to compute compared to the trivial way to compute FDRs. As a general property, the FDR can also be estimated simply by the average PEP of the significant PSMs. But conversely, the PEP is related to the derivative of the FDR which is usually inobvious. Although calculating PEP is not that easy, it is a better way to explain how good the identifications are and do further convergence analysis for different tools.
On the other hand, the PEP is strongly related to the $q$ value. For a certain PSM, its PEP value could be equal to or higher than its $q$ value, since the rate of incorrect PSMs among PSMs with a certain given score is always equal to or higher than among the set of PSMs with the same score or higher. At this point, the $q$ value for a PSM can be derived by the average of PEPs of the PSMs which have equal or higher scores.
Chapter 2

Methods

2.1 Data collection process

Firstly, the shotgun proteomics is applied to generate fragmentation spectra. Particularly, the proteins from Human Du145 prostate cancer cells were digested with trypsin. Then LC-MS/MS analysis was performed on the resulting peptides ions. The detailed experiment is described in (Serang et al., 2012). Mass spectra were acquired on an Orbitrap mass spectrometer. The datasets are available via ProteomeCommons.org Tranche.

Secondly, the isotope distributions and charge states of the analytes were determined from MS1 data using Hardklór v1.36. With use of Crux v1.39.1 and the precursor mass window was set to be ±10 ppm, the fragmentation spectra were searched against the latest human protein database (version 70), which is available via ftp://ftp.ensembl.org. For the target-decoy strategy, the datasets were also searched against a decoy database which was generated by simply reversing the protein sequences from the target database. 129 436 target scores and 129 451 decoy scores were stored in the end of this step and used for the following training and analysis. The resulting PSMs were ranked according to the SEQUEST cross-correlation score (XCorr).

Finally, the sqrt-format PSM datasets were postprocessed using Percolator v2.01 (Käll et al., 2007a). The workflow for the whole process is listed in Figure 2.1.

2.2 Chimeric MS/MS spectra

Let us recall the experimental procedure on the mass spectrometry instrument. Precursor ions of peptides observed in the MS1 spectra are isolated and broken down into small fragments, which are observed in the MS2 spectra. This leads to a complicating factor for identification, which is the problem of chimeric spectra (co-fragmentation spectra). This problem occurs when two or more precursor ions with similar mass and retention time are co-fragmented by MS/MS due to the limitation of isolation $m/z$ window width (Houel et al., 2010; Picotti et al., 2007;
Paizs and Suhai, 2005). Figure 2.2 gives a typical example of a chimeric spectrum from the data we described in section 2.1.

Depending on the mechanism of a tandem mass spectrometer, there are two analyzers stepwise producing spectra. In most cases, only MS2 spectra are used to do the matches with an existing protein database and the MS1 data is simply ignored. However, the information from MS1 spectra, such as chimeric data, influences a lot. But the influence of chimeric sequencing on the identification rate has not been explored to date. In large-scale data sets, the identification rate for chimeric MS/MS is 2-fold lower compared to non-chimeric spectra (Houel et al., 2010). In order to improve the overall rate of peptide identification, we attempted to collect the chimeric data from MS1 spectra as a auxiliary information for training the support vector machine classifier, and hope to get more improvement of the identification. The idea is also simple: for each spectrum, we calculate the fraction of the ion-current stemming from the identified peptide’s precursor ion in the corresponding isolation window.

\[
\text{feature} = \frac{\text{isotope peaks area of target}}{\text{total peaks area}}
\]  

(2.1)
Figure 2.2. Example of chimeric MS1 and MS2 spectra of the data collection from Human Du145 prostate cancer cells. 1) The upper panel shows the MS1 spectra in one isolation window with width 4 dalton (Da). \( A_0 \) and \( A_1 \) represent isotope peaks of the target precursor ion. \( B_0 \) and \( B_1 \) are isotope peaks of a contaminating precursor ion. 2) The lower panel is the MS/MS spectrum fragmented from the upper window information.

In this way, we can obtain a set of fractions for the target PSMs. For the corresponding decoy PSMs, we also assign the same values for each spectrum.
2.3 Non-parametric model to estimate PEP

As we mentioned before, PEP is usually much more difficult to compute compared to the trivial way to compute FDRs. There are several existing tools which apply parametric or semiparametric models to estimate the PEP. It has been noted by Keller (Keller et al., 2002) that we can model the PSMs as a mixture of correct and incorrect matches. Based on this way of modelling, it then leads to a regression problem of our scores. While due to the versatility of scoring function, the data is always nontrivial and then limited, then a non-parametric model could be a more general way to do so. In the following part, I would like to introduce the modeling and non-parametric regression problem as well as a new method to obtain a better fitting result.

2.3.1 Modelling

As a smart way to simulate the incorrect PSMs, let us again come back to the target-decoy strategy. After searching the data against the target database and decoy database respectively, now we are given one set of target PSM scores and one set of decoy PSM scores. But we are only interested in the PEP for each target PSM.

For a given target PSM, we call the null hypothesis $H_0$ if the target peptide is incorrectly matched, and the alternative hypothesis $H_1$ if it is correctly matched. We assume that all of our decoy PSMs are incorrect and satisfy the null hypothesis. We wish to compute the PEP for the target PSM with a score $x$:

$$ PEP(x) = Pr(H_0|Score = x). $$

Denote the random variable $X$ to be the obtained score of the PSM we are interested in and apply the Bayes rule, we have:

$$ Pr(H_0|X = x) = \frac{Pr(H_0)Pr(X = x|H_0)}{Pr(X = x)}. $$

Let $\pi_0 := Pr(H_0)$ be the prior probability of the null model, and $f_0(x) := Pr(X = x|H_0)$, $f(x) := Pr(X = x)$. We then successfully decompose the PEP into two terms: $\pi_0$ and the ratio $\frac{f_0(x)}{f(x)}$, which we will estimate separately. Firstly, $\pi_0$ is estimated using a bootstrap procedure applied on the calculated $p$-values of the decoy PSMs (Storey, 2002). Secondly, we estimate the ratio $\frac{f_0(x)}{f(x)}$ using the model based on the non-parametric logistic regression method (Green and Silverman, 1994).

Imagine if we pool all the target and decoy PSMs together and randomly pick a PSM with score $x$, then we can define $p(x)$ as the probability that the PSM we selected is from decoy database:

$$ p(x) = \frac{f_0(x)}{f(x) + f_0(x)}. $$
2.3. NON-PARAMETRIC MODEL TO ESTIMATE PEP

Hence the ratio \( \frac{f_0(x)}{f(x)} \) can be reformulated by the decoy probability \( p(x) \) as follows:

\[
\frac{f_0(x)}{f(x)} = p(x) \frac{1}{1 - p(x)}.
\]

and the problem then can be solved in an easier way by estimating this decoy probability \( p(x) \).

If we define a link function

\[
g(p) = \log\left(\frac{p}{1 - p}\right),
\]

\( g \) is the definition of the logit function of \( p \). This leads to a logistic regression problem. As we mentioned above, non-parametric regression method is utilized in this case due to the limitation of data. Following the idea of base spline and the model constructed in (Green and Silverman, 1994), we firstly divide our target and decoy PSM scores into \( N \) equal-sized bins. For each bin of scores, we record the total number of scores \( m_i \), the median score \( x_i \) and the number of decoy scores \( y_i \), \( i = 1, \ldots, N \).

Now the decoy probability \( p(x) \) has a more practical explanation if we model the three observations as the parameters and outcomes of a binomial process with the probability \( p_i = p(x_i) \). Particularly, \( Y_i \sim Bin(m_i, p_i) \):

\[
Bin(y_i|m_i, p_i) = \left( \begin{array}{c} m_i \\ y_i \end{array} \right) p_i^{y_i} (1 - p_i)^{m_i - y_i},
\]

\[
= \left( \begin{array}{c} m_i \\ y_i \end{array} \right) \exp \left[ y_i \log\left(\frac{p_i}{1 - p_i}\right) + m_i \log(1 - p_i) \right],
\]

with mean \( \mu_i = m_i p_i \) and variance \( \sigma_i = m_i p_i (1 - p_i) \). As a member of general exponential family, the binomial probabilistic density function (pdf) can be rewritten as:

\[
p(y_i | g_i) = \exp \left( y_i g_i - b(g_i) + c(y_i) \right),
\]

where \( g_i = g(x_i) \), \( b(g_i) = m_i \log(1 + e^{g_i}) \), \( c(y_i) = -\log \left( \frac{m_i}{y_i} \right) \). Note that \( b'(g_i) = \mu_i \), \( b''(g_i) = \sigma_i \).

The regression procedure will give an estimate of the linkage \( g \), yielding the desired PEP estimate as:

\[
PEP = \hat{\pi}_0 \exp \left( \hat{g}(x) \right). \tag{2.2}
\]

2.3.2 Roughness penalty approach

In non-parametric case, the regression coefficients are usually estimated using maximum likelihood estimation. Look at the Log-likelihood function of the model at first:

\[
\ell(g) = \sum_{i=1}^{N} \left[ y_i g_i - b(g_i) + c(y_i) \right].
\]
Here a roughness penalties approach is used in (Green and Silverman, 1994; Käll et al., 2008), which adds a continuous second derivative condition and leads to a Penalized log-likelihood function (PLE):

$$ PLE = \ell(g) - \frac{1}{2} \alpha \int g''(t)^2 dt, $$

where $\int g''(t)^2 dt$ is called a roughness penalty term assuring the smoothness and $\alpha$ is a regularization parameter controlling the bias-variance trade-off. To capture the optimal maximum of this PLE, they also model the link function $g(x)$ with a cubic spline.

### Cubic spline

Let $a < t_1 < \ldots < t_n < b$. A function $g$ defined on $[a, b]$ is a cubic spline if the following two conditions are satisfied:

- on each of the intervals $(a, t_1), (t_1, t_2), \ldots, (t_n, b)$, $g$ is a cubic polynomial;
- the polynomial pieces fit together at the points $t_i$ in such a way that $g, g', g''$ are continuous at each knot $t_i$.

A cubic spline on an interval $[a, b]$ is natural if its second derivatives are zero on boundary $a$ and $b$. Following the previous definition, let $g_i = g(x_i)$ and $\gamma_i = g''(x_i)$ and $h_i = t_{i+1} - t_i$ for $i = 1, \ldots, n - 1$, then

$$ g'(t^-_i) = \frac{g_i - g_{i-1}}{h_{i-1}} + \frac{1}{6} h_{i-1}(\gamma_{i-1} + 2\gamma_i), $$

$$ g'(t^+_i) = \frac{g_{i+1} - g_i}{h_i} - \frac{1}{6} h_i(\gamma_{i+1} + 2\gamma_i). $$

Note that $g'$ is continuous which implies $g'(t^-_i) = g'(t^+_i)$. Rearranging this yields

$$ \frac{g_{i+1} - g_i}{h_i} - \frac{g_i - g_{i-1}}{h_{i-1}} = \frac{1}{6} h_{i-1}\gamma_{i-1} + \frac{1}{3} (h_{i-1} + h_i)\gamma_i + \frac{1}{6} h_i\gamma_{i+1}. $$

Let $Q$ be the $n \times (n-2)$ matrix with

$$ q_{j-1,j} = h_{j-1}^{-1}, \quad q_{jj} = -h_{j-1}^{-1} - h_j^{-1}, \quad \text{and} q_{j+1,j} = h_j^{-1} $$

for $j = 2, \ldots, n - 1$, and $q_{ij} = 0$ for $|i - j| \leq 2$. We also define a $(n-2) \times (n-2)$ symmetric matrix $R$ with entries $r_{ij}$:

$$ r_{ii} = \frac{h_{i-1} + h_i}{3}, \quad i = 2, \ldots, n - 1, $$

$$ r_{i,i+1} = r_{i+1,i} = \frac{h_i}{6}, \quad i = 2, \ldots, n - 2, $$

$$ r_{i,j} = 0, \quad |i - j| \leq 2. $$
2.4. ESTIMATE PEP USING KERNEL LOGISTIC REGRESSION

Let \( g = (g_1, \ldots, g_n)^T \) and \( \gamma = (\gamma_2, \ldots, \gamma_{n-1})^T \), then vectors \( g \) and \( \gamma \) specify a natural cubic spline if and only if the condition

\[
Q^T g = R \gamma
\]

is satisfied (Green and Silverman, 1994).

Now we can focus on the expression for the roughness penalty term. Integrate by parts, use the facts that \( g''(a) = g''(b) = 0 \) and \( g''' \) is zero outside \([t_1, t_n]\), to obtain:

\[
\int_a^b g''(t)^2 dt = \sum_{i=2}^{n-1} \gamma_i \left( \frac{g_{i+1} - g_i}{h_i} - \frac{g_i - g_{i-1}}{h_{i-1}} \right)
= \gamma^T Q^T g = \gamma^T R \gamma
= g^T K g,
\]

where \( K = QR^{-1}Q^T \) is symmetric non-negative definite for approximating the quadratic integral. Furthermore, the estimation has the following form:

\[
\hat{g} = \arg \max_g PLE = \arg \max_g \left\{ \sum_{i=1}^N \left[ y_i g_i - b(g_i) \right] - \frac{1}{2} g^T K g \right\}.
\]

With use of the cubic spline strategy, the main conceptual advantage of the roughness penalty method is that it still leads to a numerical evaluation procedure for MLE due to the quadratic nature (Green and Silverman, 1994). However, the estimates rely on the calculated observations \( g_i \) that sometimes could not be a number when the denominator of fraction \( \frac{g_i - g_{i-1}}{h_{i-1}} \) is zero. When this problem occurs we may expect the results of less accuracy. This leads to the idea of using kernel logistic regression methods which avoid the direct calculation of the logit function \( g_i \), but estimate the kernel coefficients instead.

2.4 Estimate PEP using kernel logistic regression

For machine learning approach, the kernel trick is a way of mapping observations from a general set \( S \) into an inner product space \( V \) without having to compute the mapping explicitly. The trick to avoid the explicit mapping is to use learning algorithms that only require dot products between the vectors in \( V \), and choose the mapping such that these high-dimensional dot products can be computed within the original space, by means of a kernel function.

2.4.1 Mercer kernel

A Mercer kernel is a kernel which is positive semi-definite (Cawley and Talbot, 2008). Positive definiteness in the context of kernel functions implies that a matrix
created using a particular kernel is positive semi-definite as well. Note that a matrix is positive semi-definite if its eigenvalues are non-negative. When a kernel is positive semi-definite, one may exploit the kernel trick as follows.

Assume we have some mappings from an input space $S$ to a feature space $V$, then a kernel function (or kernel)

$$K(u, v) = \langle \phi(u), \phi(v) \rangle$$

can be used to define the inner product in feature space $V$. In practical applications, the squared exponential kernel, such as Gaussian kernel is most commonly used:

$$K(x, x') = \exp\left\{-\theta \|x - x'\|^2\right\} \quad \text{or} \quad K(x, x') = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2} \left(\frac{x - x'}{h}\right)^2\right\}.$$ 

### 2.4.2 Kernel logistic regression model

The kernel logistic regression (KLR) approach constructs a regression model in a high-dimensional feature space $V$ induced by a Mercer kernel:

$$g(x) = w \cdot \phi(x) + \beta,$$

where $w$ is a vector of model parameters and $\phi(\cdot)$ can be a non-linear transformation. Instead of specifying $\phi$ directly, we evaluate the inner product between the images of $x$ in the feature space by a Mercer kernel.

Now rewrite our model with use of regularized negative log-likelihood function and a vector of model parameters $w$, which leads to a minimization problem.

$$\tilde{g}_i = \arg\min_w E = \arg\min_w \{\frac{1}{2} \|w\|^2 - \sum_{i=1}^{N} [y_i g_i - b(g_i)]\}. \quad (2.4)$$

From the representer theorem (Schölkopf et al., 2001), the solution of an optimization problem of this scheme can be written in the form of a linear combination of the training patterns,

$$w = \sum_{i=1}^{N} \alpha_i \phi(x_i), \quad (2.5)$$

which implies that

$$g(x) = \sum_{i=1}^{N} \alpha_i K(x_i, x) + \beta \quad \text{and} \quad \|w\|^2 = \alpha^T K \alpha. \quad (2.6)$$

Define a convex loss function $c(g_i, y_i) = -y_i g_i + b(g_i)$ and we firstly minimize this convex loss function and then estimate the constrained optimization problem. The first and second derivatives of $c$ with respect to $g_i$ are given by

$$\frac{\partial c_i}{\partial g_i} = \mu_i - y_i \quad \text{and} \quad \frac{\partial^2 c_i}{\partial g_i^2} = \sigma_i,$$
where \( c_i = c(p_i, y_i) \). Then a weighted least-squares criterion can be substituted to locally approximate \( c_i \):

\[
q_i = \frac{\lambda_i}{2} (z_i - g_i)^2 \approx c(p_i, y_i) + \text{const}
\]

To estimate \( \lambda_i \), we need utilize the requirement that the curvature of \( q_i \) and \( c_i \) is identical at \( g_i \):

\[
\frac{\partial^2 q_i}{\partial g_i^2} = \frac{\partial^2 c_i}{\partial g_i^2} \Rightarrow \lambda_i = \sigma_i.
\]

We also require the equal gradients with respect to \( g_i \):

\[
\frac{\partial q_i}{\partial g_i} = \frac{\partial c_i}{\partial g_i} \Rightarrow -\lambda_i (z_i - g_i) = \mu_i - y_i \Rightarrow z_i = g_i + \frac{y_i - \mu_i}{\sigma_i}.
\]

Now the log-likelihood function part can be solved iteratively by updating \( \lambda_i \) and \( z_i \) in each iteration.

Note that now our optimization problem

\[
\min \tilde{E} = \frac{\gamma}{2} ||w||^2 + \frac{1}{2} \sum_{i=1}^{N} \lambda_i \epsilon_i^2
\]

subjects to

\[
g_i = w \cdot \phi(x_i) + \beta + \epsilon_i,
\]

where \( \epsilon_i = (y_i - \mu_i)/\sigma_i \). The primal Lagrangian for this optimization problem gives the unconstrained minimization problem:

\[
\min L(w, \beta, \epsilon, \alpha_i) = \frac{\gamma}{2} ||w||^2 + \frac{1}{2} \sum_{i=1}^{N} \lambda_i \epsilon_i^2 - \sum_{i=1}^{N} \alpha_i [w \cdot \phi(x_i) + \beta + \epsilon_i - z_i] \tag{2.7}
\]

To simplify our problem, the following optimality conditions are given from the expressions of derivatives.

\[
\frac{\partial L}{\partial w} = 0 \Rightarrow w = \gamma \sum_{i=1}^{N} \alpha_i \phi(x_i), \tag{2.8}
\]

\[
\frac{\partial L}{\partial \beta} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i = 0, \tag{2.9}
\]

\[
\frac{\partial L}{\partial \epsilon_i} = 0 \Rightarrow \alpha_i = \lambda_i \epsilon_i, \tag{2.10}
\]

\[
\frac{\partial L}{\partial \alpha_i} = 0 \Rightarrow w \cdot \phi(x_i) + \beta + \epsilon_i = z_i. \tag{2.11}
\]

Eliminating \( w \) and \( \epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_N) \), we can find that

\[
\gamma \sum_{j=1}^{N} \alpha_j \phi(x_j) \cdot \phi(x_i) + \beta + \frac{\alpha_i}{\lambda_i} = z_i \quad \forall i \in \{1, 2, \ldots, N\}.
\]
Furthermore, this system of linear equations can be written in matrix form as:

\[
\begin{bmatrix}
\gamma K + W & 1 \\
1^T & 0
\end{bmatrix}
\begin{bmatrix}
\alpha \\
\beta
\end{bmatrix} =
\begin{bmatrix}
z \\
0
\end{bmatrix}
\]

(2.12)

where \( K = [k_{ij} = K(x_i, x_j)]_{i,j=1}^N, W = \text{diag}\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_N^{-1}\} \) and \( \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N) \) is a vector of Lagrange multipliers.

Iteratively re-weighted least squares (IRLS)

This system can be solved iteratively since the expression of our objective function is not immediately explicit. Each iteration is a weighted least squares regression of working response vector \( z \) on the model matrix with a weights matrix \( W \) (Murphy, 2012). This therefore leads to an algorithm which is known as iteratively re-weighted least squares or IRLS for short, since we update the weight matrix \( W \) at each iteration until convergence. Both \( z \) and \( W \) are functions of the current estimate \( g \) and re-evaluated at each iteration. See Algorithm 3.

Algorithm 3 IRLS algorithm. The algorithm takes an input of the bin dataset \( G = (x, y, m) \) representing the median decoy score, number of decoy scores, total number of scores from the given set. The function \( \text{solveLS} \) applies a Cholesky decomposition procedure and the details can be found in following part. The function \( \text{kernel} \) returns a mercer kernel matrix \( K \) which is semi-positive definite.

This algorithm returns an optimal estimation of the logistic regression.

\begin{verbatim}
1: procedure KLR_IRLS(G, h)  \triangleright \ldots \text{procedure, } G = (x, y, m)
2:     \text{K} \leftarrow \text{kernel}(x, h)  \triangleright \text{Set Kernel matrix } K
3:     \alpha^0, \beta^0 \leftarrow 0  \triangleright \text{Initialize parameters}
4:     g_0 \leftarrow \text{K} \hat{\alpha}^0 + \beta^0
5:     for k \leftarrow \{1, 2, 3\} do
6:         for i \leftarrow \{1, 2, \ldots, N\} do
7:             \mu_i \leftarrow b'(g_i)
8:             \sigma_i \leftarrow b''(g_i)
9:             \lambda_i \leftarrow \sigma_i
10:            z_i \leftarrow g_i + \frac{m_i - \mu_i}{\sigma_i}
11:         end for
12:         (\hat{\alpha}^k, \hat{\beta}^k) \leftarrow \text{solveLS}(W, z, K)  \triangleright \text{build linear system of equations}
13:         g_k \leftarrow K \hat{\alpha}^k + \hat{\beta}^k
14:     end for
15:     \hat{g} \leftarrow \text{check convergence}
16:     return (\hat{g})
17: end procedure
\end{verbatim}
2.4.3 Solving linear equations

The first row of (2.12) gives:

\[(\gamma K + W)\alpha + 1\beta = z \Rightarrow \alpha = (\gamma K + W)^{-1}(z - 1\beta).\]

Using this result we can eliminate \(\alpha\) in the second row:

\[1^T(\gamma K + W)^{-1}1\beta = 1^T(\gamma K + W)^{-1}z.\]

This leads to two linear systems:

\[(\gamma K + W)\xi = 1 \quad \text{and} \quad (\gamma K + W)\zeta = z, \quad (2.13)\]

from which we can update the parameters as follows:

\[\beta = \frac{1^T\zeta}{1^T\xi} \quad \text{and} \quad \alpha = \zeta - \xi\beta. \quad (2.14)\]

Since matrix \(K\) satisfies Mercer’s theorem, \(\gamma K + W\) is then symmetric and semi-definite since \(\lambda_i = \sigma_i\) is positive. Cholesky decomposition can be applied for a more efficient implementation. Particularly, we decompose \(\gamma K + W\) as \(LL^T\) where \(L\) is lower uni-triangular matrix. The pseudocode is given in algorithm 4.

Algorithm 4 solve linear system of equations algorithm. The algorithm implements two cholesky decomposition procedures for solving the linear system. It returns the solutions of parameters \(\alpha, \beta\).

```
1: procedure SOLVE_LS(W, z, K)  \triangleright \text{procedure, solve linear system}
2: M ← γK + W \triangleright \text{Matrix M is symmetric positive definite}
3: L ← Cholesky(M) \triangleright \text{L is lower triangular matrix}
4: ξ ← solve(LL^T) = 1
5: ζ ← solve(LL^T) = z
6: β ← \frac{1^Tζ}{1^Tξ}
7: α ← ζ - ξβ
8: return (α, β)
9: end procedure
```

2.4.4 Generalized cross-validation based model selection strategy

The generalizied property of a kernel logistic regression model is however governed by a small number of parameters: \(z, \alpha, \beta, K\), the values of which must be determined during the process of model selection.

A model selection strategy for KLR was proposed by Cawley and Talbot (2008), based on a computationally efficient closed-form approximation of the leave-one-out cross-validation procedure. The idea is that the cross-validated residuals \(g_i -\)
\( \hat{g}_1^{(-i)} \) are easily related to the ordinary residuals \( g_i - \hat{g}_i \). From previous work, the parameters are iteratively determined via weighted least-squared solutions. It is suggested that leave-one-out cross-validation approach can be performed efficiently for the kernel logistic regression model. Take a close look at the first iteration of the matrix formula (2.12) as an example, decompose the parametric matrix into block matrix as follows:

\[
\begin{bmatrix}
\gamma K + W & 1 \\
1^T & 0
\end{bmatrix}
\begin{bmatrix}
c_{11} \\ c_i \\ c_1 \\ C_1
\end{bmatrix}
= C
\]  

(2.15)

Exclude the first training pattern: 

\[
[1; 1] = C_1^{-1}[z_2; \ldots; z_N, 0],
\]

and the prediction is then given by:

\[
\hat{g}_1^{(-1)} = c_i^T \alpha^{(-1)} \beta^{(-1)}
\]

\[
= c_i^T C_1^{-1}[z_2; \ldots; z_N, 0]
\]

\[
= c_i^T C_1^{-1} [c_i C_1] [\alpha^T, \beta]^T
\]

\[
= c_i^T C_1^{-1} [c_i \alpha_1 + c_1] [\alpha_2, \ldots, \alpha_N, \beta]^T.
\]

Note that, from the first equation we have:

\[
z_1 = c_1 \alpha_1 + c_i^T [\alpha_2, \ldots, \alpha_N, \beta]^T,
\]

which leads to

\[
\hat{g}_1^{(-1)} = z_1 - \alpha_1 (c_1 - c_i^T C_1^{-1} c_i).
\]

Following this work, we can formulate all the iterations by

\[
\hat{g}_i^{(-i)} = z_i - \frac{\alpha_i}{[C^{-1}]_{ii}},
\]

where \([C^{-1}]_{ii}\) is the \(ii\)-th element of the inverse matrix of \(C\).

Now the cross-validated estimate \( \hat{g}_i^{(-i)} \) is obtained in an easier computational way. For each iteration, we obtain different estimate for the validation procedure. With setting an appropriate objective function, we then can find out the best model parameters with full use of the information of data.

For any kernel methods, the choice of bandwidth affects the results most. Thus, we use this cross-validation procedure to find out the best bandwidth.

### 2.5 Accuracy of the PEP estimates

After obtaining some reasonable results, it’s always common to ask whether the performance is good. For this purpose, we used the root mean squared deviation between the PEP-derived and the FDR-derived q-value as a measurement of the quality of our PEP estimation procedure:

\[
\Delta_{rms} = \sqrt{\frac{1}{|T|} \sum_{x \in T} (q_{PEP}(x) - q_{FDR}(x))^2},
\]  

(2.16)
2.5. ACCURACY OF THE PEP ESTIMATES

Algorithm 5 The algorithm for estimating PEP by kernel logistic regression procedure. Given a set $T$ of target PSM scores and a set $D$ of a decoy PSM scores. $N$ represents the desired number of bins. The function KLR_IRLS() is used for solving the system iteratively and returns the optimal solution. The $\pi_0$.bootstrap implements the bootstrap $\pi_0$ estimation.

1: procedure KLR-ESTIMATE-PEP($T,D,N$) 
2: $A \leftarrow$ sort($T,D$) \hspace{1cm} $\triangleright$ Combine, sort target and decoy 
3: $b_1, b_2, \ldots, b_N \leftarrow$ divideIntoBins($A,N$) 
4: $G \leftarrow \{\}$ 
5: for $i \in \{1, 2, 3\}$ do 
6: $m_i \leftarrow |b_i|$ 
7: $x_i \leftarrow$ Median($b_i$) 
8: $y_i \leftarrow$ countDecoy($b_i$) 
9: $G \leftarrow G \cup \{(x_i, y_i, m_i)\}$ 
10: end for 
11: $h \leftarrow$ cross-validation($G$) \hspace{1cm} $\triangleright$ Find optimal bandwidth for kernel function 
12: $\hat{g} \leftarrow$ KLR_IRLS($G,h$) 
13: $\hat{\pi}_0 \leftarrow \pi_0$.bootstrap($A$) 
14: return $\text{pep} = \hat{\pi}_0 \exp(\hat{g})$ 
15: end procedure

where the PEP-derived q-values are computed as follows:

$$q_{PEP}(x^t) = \min_{x^t \geq x} \frac{\sum_{x \in \{y|y \geq x^t, y \in T\}} P(H_0|X = x)}{|\{y|y \geq x^t, y \in T\}|}. \quad (2.17)$$

Combining with the cross-validation procedure described in section 2.4, we use this root mean squared deviation between two types of q-values as the optimizing objective. This can make the model selection procedure more sensitive and accurate.
Chapter 3

Results

3.1 Impact of chimeric spectra

Based on the data described in Section 2.1, we look at the chimeric problem at first. For each spectrum, we calculated the fraction of the ion-current stemming from the identified peptide’s precursor ion in the corresponding isolation window. After analysis of the isolation procedure from the MS2 data, we set the isolation window width as 3.2 Dalton (Da) and extracted all the peaks within this region of the precursor ions. For a higher chimeric fraction above 0.7, we say it is non-chimeric. Setting a threshold of 0.7 for the datasets of MS/MS spectra, there were more than 50% PSMs with chimeric fraction lower than this value (Figure 3.1), which means the chimeric spectra do exist and they are quite common.

Due to this truth, we then used the chimeric fractions, together with other score features such as Xcorr, Sp scores, as the features for our support vector machine training. The re-ranked score distribution is given in Figure 3.2. Compared to the original distribution shown in Figure 1.4, the higher target scores became more distinct and the fraction of positive or correct PSMs was improved.

For the significance test, the posterior error probabilities and q-values were calculated using our kernel logistic regression method. After training, around 17 800 PSMs were accepted as significant with setting a q-value threshold 0.01. Compared to the initial figure 13 300 PSMs with $q \leq 0.01$, the improvement is dramatic. Figure 3.3 compares the number and percent of PSMs from Crux and Percolator based on the FDR-derived q-values. We also compared the effect of chimeric fraction feature and original percolator features. However, there was no big improvement as shown in the right hand side panel in Figure 3.3.

3.2 PEP estimates

Now the question is whether we have successfully identified the peptide that generated the observed spectrum. With the effort of KLR method, the PEPs were successfully estimated and presented in Figure 3.4. As a reference, the target scores
28 CHAPTER 3. RESULTS

Figure 3.1. The correlation between Xcorr scores and Chimeric fractions. As it shows, most of the PSMs with lower scores are more likely to obtain lower values of the chimeric fraction.

distribution with relative abundance was plotted in the background. This figure demonstrated the accuracy of the estimated PEPs at one point: For the PSMs with higher scores, they were less likely to be incorrect, hence they had much lower posterior error probabilities. For the lower scores below the threshold, we accepted all of them to be incorrect, hence the posterior error probability should be 1.0.

3.3 Comparison of q-values

We also plotted and compared the quantiles of two different q-value estimates in Figure 3.5. Following this work, we can take a closer look at the behavior of PSMs with q-values below threshold 0.01. In general, there is only slight difference between the two types of q-values. Since PEP-derived q-values were computed indirectly by first computing PEPs and then integrating, they were expected to be less accurate. However, for the FDR-derived q-value, it could become zero at some points in case that no decoy was simulated above some thresholds. As described in section 2.5, we also calculated the corresponding root mean squared deviation between these two measures. Associated with the cross validation strategy, we
3.4. GENERALIZED PROPERTY OF KLR METHOD

Recalling the procedure of our kernel logistic regression method, there is no specific requirement for the dataset. It is then reasonable to expect its applicability for other types of datasets.

Because of the big number of scoring algorithms and different types of instru-
Estimate PEP using kernel logistic regression by gaussian kernels

Figure 3.4. Posterior error probability for all the 129 436 target PSMs. The target score distribution with relative abundance was plotted in the background as a reference.

ments, samples, chromatography conditions in the field of mass spectrometry, a method with less assumptions should be more robust and useful. Here comes the KLR method in play. To illustrate this point, we applied our KLR procedure to other two types of datasets: the MS-GF+ scores and the correctness probability for PeptideProphet which is semiparametric (Keller et al., 2002) and already attempted to estimate the posterior probability. The comparison of results is shown in figure 3.6. The kernel logistic regression performed still very well on this MS-GF+ dataset from the left panel. The right panel presented PEP-derived q-value as a function of FDR-derived q-value for MS-GF+ input scores. Compared with the semiparametric estimates, KLR estimates still have a better correspondence with the FDR-derived q-values.

We also tested a collection of 18,149 spectra derived from a yeast whole-cell lysate. Each peptide identification method searches against the yeast proteome and produces a list of 35,676 PSMs ranked by q value, computed as described in Section 2.4. Figure 3.7 plots, for each method, the number of accepted PSMs as a function of q value threshold. We could see the KLR method still worked well by the similar analysis as above.

3.5 Software: Qvality-KLR

The software QVALITY is firstly written by Käll (Käll et al., 2008), which provides a convenient way to calculate two popular statistical confidence measures as we introduce: the q-value and the posterior error probability (PEP). The q-value is de-
3.5. SOFTWARE: QVALITY-KLR

Figure 3.5. PEP-derived q-value as a function of FDR-derived q-value.

Figure 3.6. PEP estimates based on MS-GF+ searching scores from human cancer cell data with train/test set containing 129260 positives and 129245 negatives. It separated 16225 positives in initial direction and 27359 target PSMs with q<0.0100 were found after all training procedure and based on the Kernel PEP-derived q-values. Compared to the semiparametric estimates of PEPs and corresponding q-values, 26 577 PSMs were accepted as significant (q<0.0100).

fined as the minimal false discovery rate (FDR) level at which PSMs are accepted as significant. The PEP is the probability that this score is drawn to be incorrect. PEP is sometimes also referred to be the local false discovery rate. QVALITY takes target PSM scores and decoy PSM scores as input, containing the empirical and null score distributions respectively. This program generates a list of the origi-
nal scores, q-values and PEPs as output. The input data are pooled together and binned into \(N\) equal-sized bins. This model is based on Käll’s work in (Käll et al., 2008).

Based on Algorithm 5, the program Qvality-KLR now is developed and the code is available via github ([https://github.com/percolator](https://github.com/percolator)). Computing q-values and PEPs for 130 000 scores and the same number of decoy scores takes less than 7 s on a 2.66 GHz Intel(R) 4-core processor (almost the same as Qvality).
Chapter 4

Conclusions

In this report, we present the impact of chimeric spectra on the peptide identifications. Although the improvement of significant PSMs is not obvious when we integrated the chimeric feature into post-processing tool percolator with existing features, that might caused by the high correlation between the chimeric fractions and the original database searching scores.

We further present the error rate estimates of the peptide identifications. The following several conclusions can be made based on the analysis of results: 1) The existing methods rely on parametric or semi-parametric models of the underlying score distribution as an assumption. Instead, our method relies only upon target-decoy strategy which provides a simple model of the null score distribution. So only an accompanying collection of decoy PSM scores is required. This property ensures an easy applicability. 2) Regarding the model we establish here, the good point is that the direct estimation of the ratio $f_0(x) / f(x)$ instead of estimating the two terms separately. In this way, we avoid the possible double-effective error and cut it in half. 3) Compared to general objective functions, the generalized cross-validation procedure used here is combined with the root mean squared deviation between two types of q-values as the optimizing objective. This in fact makes the choice of optimal bandwidth more sensitive and accurate. 4) The choice of bandwidth has a direct effect on the PEP estimates. The minimal root mean squared deviation is regarded as the way to find the optimal bandwidth. On the other hand, changing the value of bandwidth becomes an easy way when people expect a different number of accepted PSMs with a fixed threshold. 5) The KLR method does not make any assumptions about monotonicity of the scores. However, it is reasonable to assume that the PSM score function should be related to the underlying PEP monotonically. Therefore, a future improvement would be to consider the monotonic case separately. This weak assumption would be presumably general enough to obtain more accurate PEP estimates.
Chapter 5

Acknowledgements

I thank Viktor Granholm for reviewing this report and providing me such a nice opportunity to do this project and enter the field of Mass spectrometry-based proteomics. I also thank every one in Käll’s group at science for life laboratory: Luninnta Moruz, Magnus Rosenlund, Jorrit Boekel, Clara Higuera, Amin Saffari, for a lot of useful discussions. I specially thank my supervisor Lukas Käll for the trust, I obtained so much experience about solving problems during the project. I also thank Matthew The, Truc Vu for being my opponent. Finally I want to thank Michael Hanke in KTH and Reinhard Nabben in TU Berlin for taking care of my study in this nice Erasmus Mundus program.
Bibliography


