Optimization-Based Peptide Mass Fingerprinting for Protein Mixture Identification

Zengyou He[1], Chao Yang[1], Can Yang[1], Robert Z. Qi[2], Jason Po-Ming Tam[2], and Weichuan Yu[1] *

[1]Laboratory for Bioinformatics and Computational Biology, Department of Electronic and Computer Engineering. [2] Department of Biochemistry. The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China.

Motivation: In current proteome research, peptide sequencing method is probably the most widely used method for protein mixture identification. However, this peptide-centric method has its own disadvantages such as the immense volume of tandem Mass Spectrometry (MS) data for sequencing peptides. With the fast development of technology, it is possible to investigate other alternative techniques.

Peptide Mass Fingerprinting (PMF) has been widely used to identify single purified proteins for more than 15 years. Unfortunately, this technique is less accurate than peptide sequencing method and can not handle protein mixtures, which hampers the widespread use of PMF technique. If we can remove these limitations, PMF will become a useful tool in protein mixture identification.

Results: We first formulate the problem of PMF protein mixture identification as an optimization problem. Then, we show that the use of some simple heuristics enables us to find good solutions. As a result, we obtain much better identification results than previous methods. Moreover, the result on real MS data can be comparable with that of the peptide sequencing method. Through a comprehensive simulation study, we identify a set of limiting factors that hinder the performance of PMF method in protein mixtures. We argue that it is feasible to remove these limitations and PMF can be a powerful tool in the analysis of protein mixtures.

Availability: The source codes in Java and the data sets are available at http://bioinformatics.ust.hk/PMFMixture.rar.

Contact: eeyu@ust.hk

1 INTRODUCTION

The identification and quantification of proteins expressed in a cell or tissue is an explicit goal of proteomics. Among existing protein identification strategies, peptide mass fingerprinting (PMF) has been widely used to identify single purified proteins since 1993 (James *et al*., 1993; Mann *et al*., 1993; Pappin *et al*., 1993; Yates *et al*., 1993). However, PMF has begun to fall out of favor in much of proteomics community (Yang *et al*., 2008) because of recent advances in the analysis of complex protein mixtures using shotgun proteomics (Aebersold and Mann, 2003; Link *et al*., 1999; Gygi *et al*., 1999; Washburn *et al*., 2001). The shotgun proteomic strategy combines protein digestion and tandem MS (MS/MS) based peptide sequencing to perform peptide-centric identification.

The PMF method has two inherent disadvantages: (1) it is less accurate than peptide sequencing method since it can't distinguish different peptides with identical mass; (2) it is originally designed for identifying single purified proteins rather than protein mixtures. If we can overcome these limitations, there will be great potential to use PMF method as an alternative or supplement of the peptide sequencing method in analyzing complex protein mixtures.

To date, some effective PMF scoring and searching methods such as MASCOT (Perkins *et al*., 1999), MS-Fit (Clauser *et al*., 1999) and ProFound (Zhang and Chait, 2000) have already been widely used. Meanwhile, more PMF algorithms are developed to further improve the identification accuracy (Margnin *et al*., 2004; Siepen *et al*., 2007; Yang *et al*., 2008). As shown by Yang *et al*. (2008), PMF method has the potential of rivaling the accuracy of peptide sequencing method.

The use of PMF method in protein mixtures has been studied in (Jensen *et al*., 1997; Park and Russell, 2001; Eriksson and Fenyö, 2005; Lu *et al*., 2008). A *subtraction* strategy is proposed by Jensen *et al*. (1997) in which the masses matching the protein identified in the first round are removed prior to the second round of searching. This procedure repeats a number of rounds until enough proteins are identified. The same strategy is also adopted in (Park and Russell, 2001; Eriksson and Fenyö, 2005). Though the *subtraction* approach can be used to identify component proteins from simple mixtures, it still suffers from the problem of random matching and has poor performance when the mixture is complex and noisy. The approach proposed in (Lu *et al*., 2008) relies on a randomized decoy database and high mass accuracy capability of mass spectrometry. From the viewpoint of PMF ranking, this method is still a traditional PMF method in which each single protein is ranked separately.

The main objective of this paper is to study the potentials and pitfalls of PMF method in protein mixtures. More concisely, we are interested in answering the following two questions:

(1) Is it possible to achieve an acceptable identification accuracy in protein mixtures if we only use PMF method on single MS data? The answer to this question is *yes*. We first show that the PMF searching in protein mixtures is actually a mathematical optimization problem. While obtaining the optimal solution is difficult, we are still able to employ some heuristic searching methods to find the local optimum. As a result, the identification performance reaches an acceptable level and it is significantly better than that of traditional PMF method and *subtraction* strategy. This demonstrates that PMF method has the potential of competing with peptide sequencing method in protein mixture identification when advanced computational methods are exploited.

(2) What are the limiting factors in the use of PMF method in protein mixture identification? Through a comprehensive simulation study, we show that the performance of PMF is mainly affected by the mass accuracy of mass spectrometer, the number of component

*to whom correspondence should be addressed proteins in the mixture, the sequence coverage of each protein and the noise level in MS data. With further improvements in MS instrumentation, protein and peptide separation techniques and computational data analysis tools, it will likely become possible to overcome these limitations. Potentially, PMF will become a powerful tool in the analysis of protein mixtures.

The rest of the paper is organized as follows: Section 2 describes the problem formulation and introduces two new PMF algorithms for identifying proteins from mixtures. Section 3 shows the potential of PMF algorithm in the analysis of protein mixtures and investigates some major limiting factors through a carefully designed experimental study. Section 4 presents some discussions. Section 5 concludes this paper.

2 METHODS

2.1 PMF for Single Protein Identification

Traditionally, PMF for single protein identification consists of the following steps:

(1) Protein purification: 2D gel-based separation produces purified protein samples.

(2) Protein digestion: Protease (such as trypsin) digests each protein into peptide mixtures and MS records the masses of resulting peptides.

(3) Protein identification: PMF scoring function ranks each protein in the database according to its match quality with the MS spectrum and reports best ones to the users.

The key point in PMF is to define a good scoring function so that the ground-truth protein is given a high rank. Most existing PMF algorithms require a user-specified mass tolerance threshold as input to define the peak matching relationship. The underlying assumption is that one experimental peak is considered corresponding to a theoretical peak if their distance is not larger than the mass tolerance threshold. Suppose we have a set of experimental peaks $Z = (z_1, z_2, ...z_l)$ and a database of proteins $D = (X_1, X_2, ...X_g)$, where X_i denotes an individual protein and g denotes the size of the database. The quality of matching between a protein $X_i \in D$ and the experimental peak set Z is measured by a scoring function $S^{(L)}(Z, X_i, \sigma)$, where σ is the mass tolerance threshold. When σ is fixed, we may use $S^{(L)}(Z, X_i)$ instead of $S^{(L)}(Z, X_i, \sigma)$.

If we assume that the ground-truth protein has the highest score, then the problem of single protein identification is actually an optimization problem:

$$\widehat{X} = \arg\max_{X_i \in D} S^{(L)}(Z, X_i), \tag{1}$$

where $\widehat{X}$ is the protein that achieves the highest score among all proteins in the database. In other words, $\widehat{X}$ best "explains" Z.

2.2 PMF for Protein Mixture Identification

From the viewpoint of data analysis, we have the same input in the context of protein mixtures: a set of experimental peaks $Z = (z_1, z_2, ...z_n)$ generated from peptide mixtures of multiple unknown proteins. Here our objective is to find a set of proteins $\widehat{Y}$ that best "explain" Z:

$$\widehat{Y} = \arg\max_{Y \subseteq D} S^{(M)}(Z, Y), \tag{2}$$

where Y denotes a subset of proteins and $S^{(M)}(Z, Y)$ is a scoring function that measures the quality of matching between Z and Y. Note that $S^{(M)}(\cdot,\cdot)$ is different from $S^{(L)}(\cdot,\cdot)$ as Y may have multiple proteins. The definition of $S^{(M)}(\cdot,\cdot)$ will be discussed in the next section.

Obviously, single protein identification is a special case of protein mixture identification when an additional constraint $|Y| = 1$ is provided.

After formulating the problem of protein mixture identification as a generic optimization problem, our task becomes: (1) defining the objective function (scoring function) $S^{(M)}(Z, Y)$ and (2) finding a good solution to the problem.

2.3 Scoring Function for Protein Mixture Identification

2.3.1 Generic Scoring Function

To define $S^{(M)}(Z, Y)$, we have two natural choices:

(1) Virtual single protein approach: We can regard Y as a "virtual" single protein V whose digestion result is the set of peptides generated from proteins in Y. Then, the score for this virtual protein is calculated in the same way as for single protein:

$$S^{(M)}(Z, Y) = S^{(L)}(Z, V). \tag{3}$$

(2) Peak partition approach: The basic idea is to distribute peaks to different proteins in Y explicitly. If we assume that each peak can only be assigned to one protein, then we need to partition the peaks of Z into disjoint subsets[1]. Without loss of generality, we assume that Y consists of k proteins $X_{s_1}, X_{s_2}, ..., X_{s_k} (1 \le s_j \le g)$ and divide Z into k disjoint subsets $Z_1, Z_2, ..., Z_k$. Then the score is calculated as:

$$S^{(M)}(Z, Y) = \sum_{j=1}^{k} S^{(L)}(Z_j, X_{s_j}). \tag{4}$$

When the single protein identification method is applied directly to protein mixtures, the scoring function is actually $S^{(L)}(Z, X_{s_j})$ in which protein X_{s_j} is used to match/explain all the peaks in Z. Clearly, this scheme will suffer from serious random matching.

The *subtraction* strategy (Jensen *et al.*, 1997) can be regarded as a special instance of peak partition approach in which peaks are divided in a greedy manner. Suppose the peak subsets $Z_0, Z_1, Z_2, ..., Z_k$ are generated by *subtraction* strategy sequentially (Z_0 is an empty set), then the score at each step $j (1 \le j \le k)$ is calculated as:

$$S^{(L)}(Z - \bigcup_{t=0}^{j-1} Z_t, X_{s_j}). \tag{5}$$

Although the *subtraction* strategy partitions the peaks into different groups, it evaluates each protein using a much larger

[1] In general, such assumption is invalid since one peak may belong to multiple proteins. A more practical model is to allow overlaps between different subsets. Further investigation towards this direction is beyond the scope of this paper.

subset of Z instead of its corresponding peak subset. Obviously, this strategy still suffers from the problem of random matching.

In short, existing scoring functions are ill posed for protein mixture identification and the development of new scoring functions is necessary.

In addition to the virtual single protein approach and peak partition approach, we can also design other kinds of scoring functions for protein mixtures. However, further study on this topic is beyond the scope of this paper since our main objective is to demonstrate the potential of PMF method in protein mixture identification when advanced computational methods are used.

Here we use the virtual single protein approach to define the scoring function. This approach has the following advantages: (1) it is simple to understand and easy to implement; (2) the calculation of score can be very efficient in the optimization process if the single protein scoring function is properly selected. Overall, it should suffice for illustrating the benefit of optimization-based PMF method for protein mixture identification.

2.3.2 Materialized Scoring Function

In our implementation, we can choose any scoring function such as the popular Mascot[2] and ProFound. Here, we use the scoring function proposed in (Samuelsson et al., 2004) due to the following reasons:

1. It has comparable performance to Mascot and ProFound.
2. Its score can be calculated quickly and incrementally in the context of protein mixtures.

The empirical results in (Samuelsson et al., 2004) support the first point and we will discuss the second point in the next section.

The basic idea of Samuelsson et al. (2004) is to consider a good match as something unlikely to happen. If one protein X_i matches r_i peaks in Z, the algorithm computes *a priori* random probability that these r_i matches occur. The score is taken as the negative logarithm of that probability. A high score value reflects an unlikely event, and hence a high degree of good matching.

There are different ways to compute the *a priori* random probability. One widely used strategy is to apply the binomial distribution (Berndt et al., 1999; Wool and Smilansky, 2002; Samuelsson et al., 2004; Eriksson and Fenyö, 2004). Then, the probability that a protein X_i has r_i random matched peaks in Z is given by:

$$Pr(|M_Z(X_i)| = r_i) = C_l^{r_i} p_i^{r_i} (1 - p_i)^{l - r_i}, \qquad (6)$$

where $M_Z(X_i)$ denotes a subset of Z whose peaks match protein X_i, l is the number of observed peaks in Z, and p_i is the probability for at least one match between a peak from Z and one of the n_i peptide masses of protein X_i.

The value of p_i is calculated as (Samuelsson et al., 2004):

$$p_i = 1 - (1 - 2\sigma/\Delta)^{n_i}, \qquad (7)$$

where Δ is the acquisition mass range (i.e. the difference between the maximum and minimum mass values of Z) and σ is the mass tolerance threshold.

The interpretation of p_i is straightforward: If we assume the probability of random match for an observed peak is $2\sigma/\Delta$, then the probability for a random miss is $1 - 2\sigma/\Delta$. If we draw n_i random peptides, then the probability of missing this observed peak in all n_i trials is $(1 - 2\sigma/\Delta)^{n_i}$. Therefore, the probability for at least one match is $1 - (1 - 2\sigma/\Delta)^{n_i}$.

The scoring function is defined as the negative natural logarithm of $Pr(|M_Z(X_i)| = r_i)$:

$$S^{(L)}(Z, X_i) = -\ln C_l^{r_i} - r_i \ln p_i - (l - r_i)\ln(1 - p_i). \qquad (8)$$

Suppose Y consists of k proteins $X_{s_1}, X_{s_2}, ..., X_{s_k}$, we can consider Y as a virtual single protein and the generalized scoring function becomes:

$$S^{(M)}(Z, Y) = -\ln C_l^{r_Y} - r_Y \ln p_Y - (l - r_Y)\ln(1 - p_Y), \quad (9)$$

where $r_Y = |\bigcup_{j=1}^{k} M_Z(X_{s_j})|$ and $p_Y = 1 - (1 - 2\sigma/\Delta)^{\sum_{j=1}^{k} n_{s_j}}$.

2.4 Local Search Algorithms for Protein Mixture Identification

After defining the scoring function for protein mixture identification, we need to solve the optimization problem by finding a subset of proteins that maximizes the objective(scoring) function. In our study, we first assume that the true number of ground-truth proteins is known in advance, i.e. k is an input parameter. Consequently, we will relax this requirement by introducing an adaptive algorithm that can determine the number of target proteins automatically.

Even when k is given, an exhaustive search is prohibitive since there are totally C_g^k possible solutions, where g is the size of protein sequence database. A variety of well known searching techniques, including simulated annealing and genetic algorithms, can be applied to find a reasonable solution. As we plan to use large protein database, computationally expensive approaches become unattractive. Here we use local-search heuristics to find good solutions efficiently.

2.4.1 Local Search Algorithm with Known k

This section presents a local search algorithm with known k for protein mixtures. We name it Losak and describe the detail in Algorithm 1.

The Losak algorithm takes the number of target proteins as input and iteratively improves the value of objective function. Initially, we randomly select k proteins and label them as "target" proteins. In the iteration process, for each protein labeled as "non-target" protein, its label is exchanged with each of the k target proteins and the objective value is re-evaluated. If the objective value increases, the protein's "non-target" label is exchanged with the "target" label of the protein that achieves the best objective value and the algorithm proceeds to the next protein. When all "non-target" proteins have been checked for possible improvements, a full iteration is completed. If at least one label has been changed in one iteration, we initiate a new iteration. The algorithm terminates when a full iteration does not change any labels, thereby indicating that a local optimum is reached.

In this algorithm, the key step is how to efficiently calculate the new score when two proteins are swapped. Thanks to the good

[2] The technical details of Mascot are not publicly available, making it difficult to apply this method directly to protein mixture identification.

Algorithm 1 Losak Algorithm

```
Input:
$D$: database of $g$ proteins; $Z$: observed peak list.
$\sigma$: mass tolerance threshold; $k$: number of target proteins.
Output:
$Y$: a set of $k$ proteins

/* Phase 1-Initialization */
Randomly select $k$ proteins into $Y$ as "target" proteins

/* Phase 2-Iteration */
Initialize $hasSwap \leftarrow$ True
while $hasSwap$=True do
    $hasSwap \leftarrow$ False
    for $i = 1$ to $g$ do
        if $X_i$ does not belong to $Y$ then
            $h \leftarrow \arg\max_j S^{(M)}(Z, Y + \{X_i\} - \{X_{s_j}\})$
            if $S^{(M)}(Z, Y + \{X_i\} - \{X_{s_h}\}) > S^{(M)}(Z, Y)$ then
                Update $Y$ as $Y \leftarrow Y + \{X_i\} - \{X_{s_h}\}$
                $hasSwap \leftarrow$ True
            end if
        end if
    end for
end while
return $Y$
```

property of scoring function in Equation (9), we can calculate the score incrementally:

When X_{s_h} in Y is swapped out and X_i is swapped in, the calculation of new p_Y is very efficient since we can store $\sum_{j=1}^{k} n_{s_j}$ as a constant C so that $p_Y = 1 - (1 - 2\sigma/\Delta)^{C - n_{s_h} + n_i}$.

In order to calculate r_Y efficiently, we use an integer array A of length l to record the number of proteins in Y that hit each observed peak. Given A, r_Y is calculated as the number of non-zero entries of A. When X_{s_h} is exchanged with X_i, we just need to use the non-overlapping elements of $M_Z(X_i)$ and $M_Z(X_{s_h})$ to update A and calculate the new r_Y value. Therefore, r_Y can be computed incrementally with a time complexity of $O(l)$.

After obtaining the values of p_Y and r_Y, we can calculate the score immediately according to Equation (9). Hence, the time complexity of incremental score calculation at each swap evaluation step is $O(l)$. Accordingly, the Losak algorithm has a time complexity of $O(qgkl)$, where q is the number of iterations, g is the number of proteins in the database, k is the number of target proteins and l is the number of observed peaks. Obviously, the Losak algorithm has good scalability since its time complexity is linear to all major parameters.

2.4.2 *Local Search Algorithm with Unknown k*

In the Losak algorithm, we assume that the number of target proteins is known based on prior knowledge. Unfortunately, such information is often not available. Thus, we need an algorithm that can determine k automatically. This section presents the Losau algorithm (Algorithm 2), which is an adaptive local search algorithm with unknown k for protein mixtures.

This algorithm is an extension of Losak algorithm and provides several additional salient features:

(1) It is adaptive: To determine the number of target proteins automatically, we introduce the "insert" operation and "delete" operation into the local optimization process so that k can be varied. If protein X_i is contained in Y, we will delete it from Y if such an operation increases the score. Similarly, if protein X_i is not contained in Y, we will either insert it into Y or exchange it with another protein in Y if such an operation increases the score.

(2) It follows the Occam's razor principle (Blumer et al., 1987). Occam's razor, also known as the *principle of parsimony*, has many applications in different areas. Often, it is phrased as "all other things being equal, the simplest solution is the best". To apply the Occam's razor principle, we need to properly interpret the meaning of "simplicity", which is usually called Occam simplicity. In our context, we can use the number of target proteins in Y as the Occam simplicity measure. Therefore, we introduce a penalty ω on the insert operation to reflect our intention of "explaining" Z using as few number of proteins as possible.

In the first iteration, we have to assign a proper value to ω so that Y can be expanded to a reasonable size. If ω is too small, the size of Y will increase too quickly to destroy the Occam simplicity. If ω is too large, the size of Y will not exceed its initial value of two and the algorithm will terminate with a poor solution. Here we use the number of proteins in Y to update the ω value in the first iteration. This initialization method has good performance in practice.

In the consequent iterations, we also need to adjust ω value to achieve a good simplicity-quality tradeoff. Since the score increases strictly at each iteration, we just need to decrease the ω value accordingly. Here we use the parameter $df (0 < df < 1)$ as the decay factor to decrease ω at each iteration: $\omega \leftarrow df \cdot \omega$.

Generally, a small df value provides us the potential of obtaining more true positives but also more false positives. In practice, we prefer a small df value since we can remove those unwanted proteins using the filtering method introduced below.

(3) It has a filtering mechanism to remove false positives effectively. The adaptive nature of Losau algorithm may introduce many false positives into the final protein list. It is desirable to filter out these incorrect proteins from the result. Meanwhile, it is also necessary to provide a significance-test-alike procedure to evaluate the confidence of each single protein. Algorithm 3 describes our filtering procedure.

In this algorithm, we evaluate each protein X_{s_j} using the peak subset $M_Z(X_{s_j})$. The idea is very simple: since all the peaks in $M_Z(X_{s_j})$ match X_{s_j}, the probability that other proteins in database achieve better single protein identification score than X_{s_j} on $M_Z(X_{s_j})$ is very low if X_{s_j} is the ground-truth protein. We use the number of "winning proteins" to measure the rank uncertainty and θ as the threshold. If there are more than θ proteins outperform X_{s_j} on $M_Z(X_{s_j})$ in terms of single protein identification score, we remove it from the result set. In general, θ value determines the filtering percentage. In the algorithm, we set $\theta = 1$ or $\theta = 2$ and we find that such parameter setting works well.

Note that we can also apply the filtering procedure to Losak algorithm in the same way, while the performance gain is not as significant as that of Losau algorithm.

(4) It has the same time complexity as Losak. In a similar way, we can show that the time complexity of incremental score calculation when "insert" operation and "delete" operation are invoked is also $O(l)$. If there are at most k proteins contained in Y in the intermediate steps, then the time complexity of Losau algorithm is still $O(qgkl)$.

Algorithm 2 Losau Algorithm

```
Input:
$D$: database of $g$ proteins; $Z$: observed peak list.
$\sigma$: mass tolerance threshold; $df$: decay factor.
$\theta$: rank threshold in filtering.
Output:
$Y$: a set of $k$ proteins. /* $k$ is determined automatically */

/* Phase 1-Initialization */
Randomly select 2 proteins into $Y$ as "target" proteins
Initialize $\omega \leftarrow 0$ and $iter \leftarrow 1$ /* $\omega$: penalty */

/* Phase 2-Iteration */
Initialize $hasOperation \leftarrow$ True
while $hasOperation$=True do
  $hasOperation \leftarrow$ False
  if $iter > 1$ then
    $\omega \leftarrow df \cdot \omega$
  end if
  for $i = 1$ to $g$ do
    $\zeta_{noop} \leftarrow S^{(M)}(Z, Y)$
    if $iter = 1$ then
      $\omega \leftarrow |Y|$
    end if
    if $X_i$ does not belong to $Y$ then
      $h \leftarrow \arg\max_j S^{(M)}(Z, Y + \{X_i\} - \{X_{s_j}\})$
      $\zeta_{swap} \leftarrow S^{(M)}(Z, Y + \{X_i\} - \{X_{s_h}\})$
      $\zeta_{inst} \leftarrow S^{(M)}(Z, Y + \{X_i\}) - \omega$
      if $\zeta_{swap} > \zeta_{inst}$ and $\zeta_{swap} > \zeta_{noop}$ then
        /* Swap Operation */
        Update $Y$ as $Y \leftarrow Y + \{X_i\} - \{X_{s_h}\}$
        $hasOperation \leftarrow$ True
      end if
      if $\zeta_{inst} > \zeta_{swap}$ and $\zeta_{inst} > \zeta_{noop}$ then
        /* Insert Operation */
        Update $Y$ as $Y \leftarrow Y + \{X_i\}$
        $hasOperation \leftarrow$ True
      end if
    end if
    if $X_i$ belongs to $Y$ then
      if $S^{(M)}(Z, Y - \{X_i\}) > \zeta_{noop}$ then
        /* Delete Operation */
        Update $Y$ as $Y \leftarrow Y - \{X_i\}$
        $hasOperation \leftarrow$ True
      end if
    end if
  end for
  $iter \leftarrow iter + 1$
end while

/* Phase 3-Filtering (See Algorithm 3) */
$Y \leftarrow ProteinFilter(D, Z, \sigma, \theta, Y)$
return $Y$
```

Algorithm 3 PoteinFilter Algorithm

```
Input:
$D, Z, \sigma, \theta$.
$Y$: a set of proteins.
Output:
$F$: a refined set of proteins, $F \subseteq Y$.

Initialize $F \leftarrow \emptyset$
for $j = 1$ to $|Y|$ do
  Initialize $Winner \leftarrow 0$
  for $i = 1$ to $g$ do
    if $S^{(L)}(M_Z(X_{s_j}), X_i) > S^{(L)}(M_Z(X_{s_j}), X_{s_j})$ then
      $Winner = Winner + 1$
    end if
  end for
  if $Winner < \theta$ then
    Update $F$ as $F \leftarrow F + \{X_{s_j}\}$
  end if
end for
return $F$
```

2.4.3 The Convergence of Two Algorithms

The convergence of the proposed algorithms is described in Theorem 1 below. With the formal proofs, we assure that these algorithms can be used safely.

Theorem 1. Both Losak algorithm and Losau algorithm converge to a local maximal solution in a finite number of iterations.

Proof. We first note that there are only a finite number (2^g) of possible subset of D. We then note that each possible subset Y appears at most once during the iterations since the sequence $S^{(M)}(\cdot, \cdot)$ is strictly increasing in both algorithms. Hence, the result follows.

Theoretically, the algorithms will converge to different local optimal solutions using different initializations. In practice, we observed that different starting points usually lead to quite similar results. This fact indicates that our algorithms are robust with respect to the initialization of starting points.

3 EXPERIMENTS

We use both simulation data and real data to demonstrate the superiority of our algorithms and the potential of PMF based protein mixture identification. We also empirically identify some major factors that affect the performance of PMF methods in protein mixtures through simulation study.

3.1 Evaluation Criteria and PMF Algorithms

Since we know the ground-truth proteins in both simulation and real data, each protein in the final protein list Y is counted as a true positive (TP) if it belongs to ground-truth proteins or as a false positive (FP) otherwise. Then, we can use standard performance metrics in information retrieval, including $precision$, $recall$, and $F1$-$measure$, to evaluate the identification performance. Their definitions are given as follows:

- n_{TP}: the number of true positives.
- n_{FP}: the number of false positives.
- n_P: the number of all ground-truth proteins.
- $precision = n_{TP}/(n_{TP} + n_{FP})$, the proportion of identified ground-truth proteins to all identified proteins.
- $recall = n_{TP}/n_P$, the proportion of identified ground-truth proteins to all ground-truth proteins.

- $F1\text{-}measure = \frac{2 \cdot precision \cdot recall}{precision + recall}$, the harmonic mean of *recall* and *precision*.

In evaluation, we compare the following algorithms:

(1) SPA: single protein identification algorithm with Equation (8) as the scoring function.

(2) Subtraction algorithm with Equation (8) as the scoring function in each step.

(3) Losak algorithm.

(4) Losau algorithm.

In the experiments, we let SPA algorithm, Subtraction algorithm, and Losak algorithm to report n_P proteins since we know the exact number of ground-truth proteins. In this setting, the $precision$, $recall$ and $F1\text{-}measure$ of each algorithm are identical. In the Losau algorithm, these performance metrics will have different values since it adaptively determines the number of target proteins. Thus, we need to report the $precision$, $recall$ and $F1\text{-}measure$ of Losau algorithm, respectively.

Throughout the experiments, we use the same set of parameters in all PMF algorithms: trypsin digestion with a maximum of one missed cleavage, monoisotopic peaks, single charge state, unrestricted protein mass. Other parameter specification will be given at the right time.

In Losau algorithm, the *df* is fixed to 0.9 and θ is fixed to 2. In both Losak and Losau algorithm, only proteins in the database that match at least five peaks are considered as potential candidates in the local optimization process.

3.2 Simulation Study

3.2.1 Simulator

We use the following procedure to generate synthetic protein mixture data, which has also been exploited in (Eriksson and Fenyö, 2005, 2007).

Firstly, we randomly select a set of proteins from the sequence database (Swiss-Prot, Release 52) as the ground-truth proteins. To ensure that each ground-truth protein has a reasonable number of digested peptides, we restrict the molecular weight of each protein between 30,000 Da and 100,000 Da.

Secondly, we perform trypsin-based protein digestion in silico (1 missed cleavage sites) and simulate the peptide detectability by retaining only a portion of proteolytic peptides according to the *sequence coverage* parameter. Here we define the *sequence coverage* as the ratio between the number of detectable peptides and the number of all peptides within the mass acquisition range (800-4500 Da in our simulation).

Finally, we simulate the experimental mass error and noise. We alter the mass of each peptide by adding a number randomly generated from a Gaussian distribution (standard deviation = σ, i.e. the mass tolerance threshold). We also add a set of noisy peaks that are randomly generated and uniformly distributed within the mass acquisition range. We determine the number of noisy peaks using the *noise level* parameter, which is defined as the ratio between the number of man-made noisy peaks and the number of total peaks (after adding these noisy peaks).

3.2.2 The Effect of Mass Accuracy

To test how the identification results of different methods are affected by mass accuracy, we generate simulation data with mass error (σ) of 0.01, 0.02, 0.03 and 0.04 Da, respectively. Each simulation data with specific mass error contains 10 mixtures of proteins. In each mixture, the number of ground-truth proteins is 20, the sequence converge is 0.3, and the noise level is 50%.

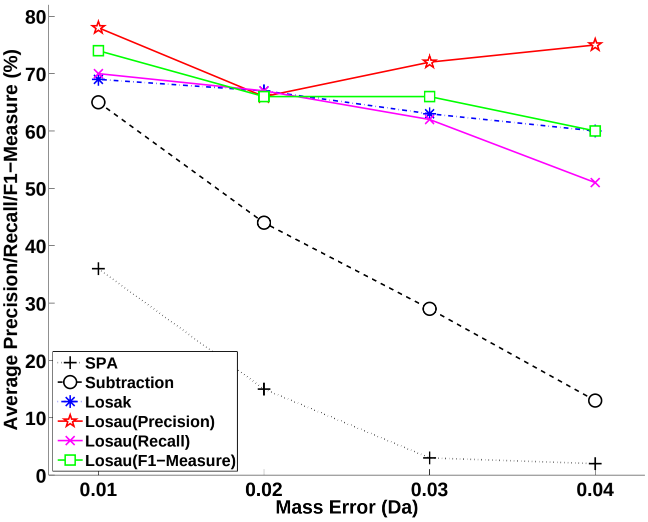

Fig. 1. Identification performance comparison for the mass error between 0.01 and 0.04 Da.

In all PMF methods, the mass tolerance threshold is set to be the known mass error. The average $precision/recall/F1\text{-}measure$ at each mass error over 10 protein mixtures are used to compare different methods, as shown in Fig.1.

The increase of mass error will decrease the identification performance of various PMF algorithms. This general trend indicates that high mass accuracy is a necessary condition for the success of PMF method in protein mixture identification.

Our algorithms are significant better than other algorithms. Moreover, our algorithms are very robust to the increase of mass error and can produce good results when the mass error is relatively larger. In contrast, the Subtraction algorithm requires smaller mass error to achieve comparable performance.

The mass error can also affect the running time of our PMF methods. Fig.2 shows that running time of our algorithms increases with the increase of mass error. This is because we use the mass error as the mass tolerance threshold. A large mass tolerance threshold will introduce more candidate proteins so that the potential search space of our algorithms is enlarged. Therefore, our algorithms need more running time to complete database searching. In contrast, SPA and Subtraction are less affected.

We can see that the running time of Losak increases linearly with respect to mass error. However, the running time of Losau increases dramatically when mass error is raised from 0.02 to 0.03 and thereafter begins to drop down. It indicates two facts: (1) Losak is more stable than Losau with respect to mass error and (2) Losau achieves comparable identification performance at the expense of much more computational time.

To illustrate the difference of running time between Losak and Losau, we plot the number of iterations used by both algorithms in Fig.3. Clearly, Losau always uses more iterations than Losak. Since the time complexity of both Losak and Losau is proportional

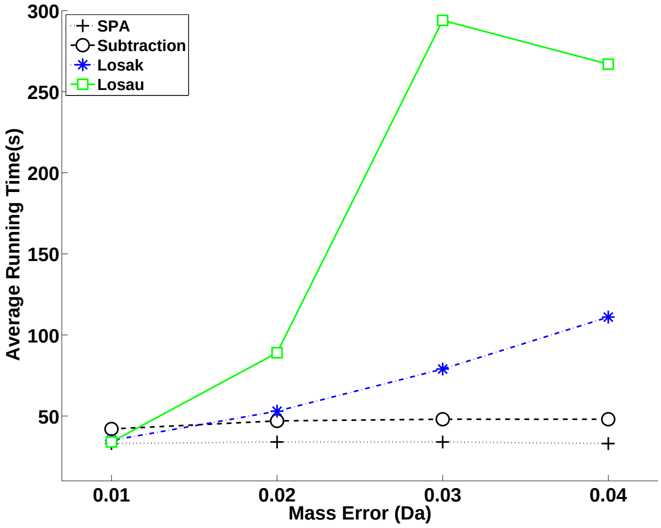

Fig. 2. Running time comparison for the mass error between 0.01 and 0.04 Da.

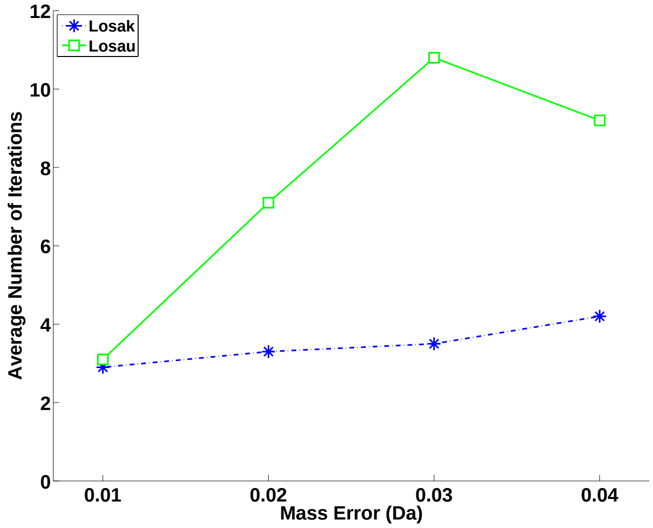

Fig. 3. Comparison of Losak and Losau in terms of the number of iterations for the mass error between 0.01 and 0.04 Da.

to the number of iterations, this plot provides a vivid example of complexity comparison.

It should be noted that the complexity of Losau is not completely predictable, as shown in the decrease of computational time at mass error of 0.04 Da. We believe the reason lies in the initial settings of the local-search method, which is common in such heuristic-based methods.

3.2.3 The Effect of Sequence Coverage

Sequence coverage is the ratio between the number of detectable peptides and the number of all peptides within the mass acquisition range. Obtaining sufficient sequence coverage is of primary importance in the context of PMF. We generate simulation data

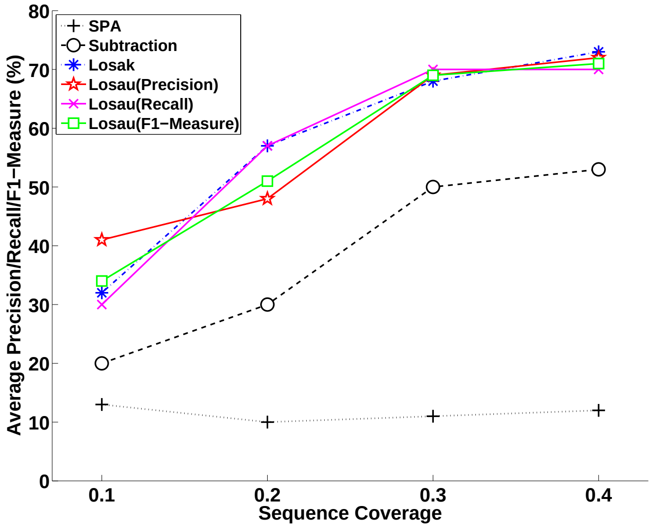

Fig. 4. Identification performance comparison for the sequence coverage ranging from 0.1 to 0.4.

with sequence coverage 0.1, 0.2, 0.3 and 0.4, respectively. At each specific sequence coverage value, we generate 10 synthetic protein mixtures using the following parameters: 20 proteins, mass error 0.02 Da and 50% noisy peaks.

Fig.4 shows that high sequence coverage is a necessary condition to accurately identify component proteins from mixtures using PMF. When the sequence coverage is low (e.g. 0.1), all algorithms perform poorly. When the coverage ratio increases, our algorithms begin to beat other methods significantly.

Fig.5 shows that the running time of different PMF algorithms increases when the sequence coverage increases. Since a high sequence coverage will introduce more peaks into the mixture data, PMF algorithms need more running time to perform protein-peak matching.

In this experiment, the running time of Losak and Losau increases with respect to the sequence coverage. On average, Losau needs more time than Losak. Fig.6 depicts the number of iterations used by Losak and Losau, which shows that Losau is more time-consuming because it needs more iterations to converge.

3.2.4 The Effect of Noise Level

MS data is very noisy. Due to various reasons, the input peak list for PMF contains many noisy peaks corresponding to chemical impurities or other components. Even in spectra generated from a single protein, there are usually more than 50% noisy peaks. Therefore, the ability to identify proteins from noisy mixtures is absolutely indispensable.

We set the noise level to 10%, 30%, 50% and 70%, respectively. At each noise level, we generate 10 mixtures using the following parameters: 20 proteins, mass error 0.02 Da and sequence coverage 0.3.

Fig.7 plots the identification results of different methods when the noise level ranges from 10% to 70%. The performance of Subtraction algorithm and SPA algorithm decline significantly when more noisy peaks are included, while our algorithms are very robust

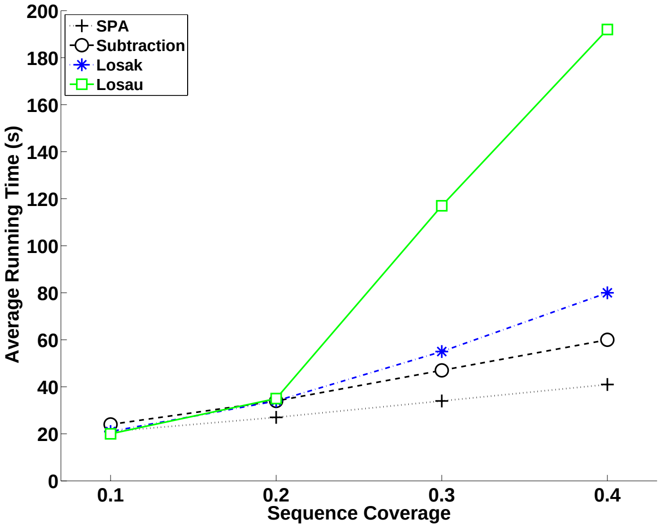

Fig. 5. Running time comparison for the sequence coverage ranging from 0.1 to 0.4.

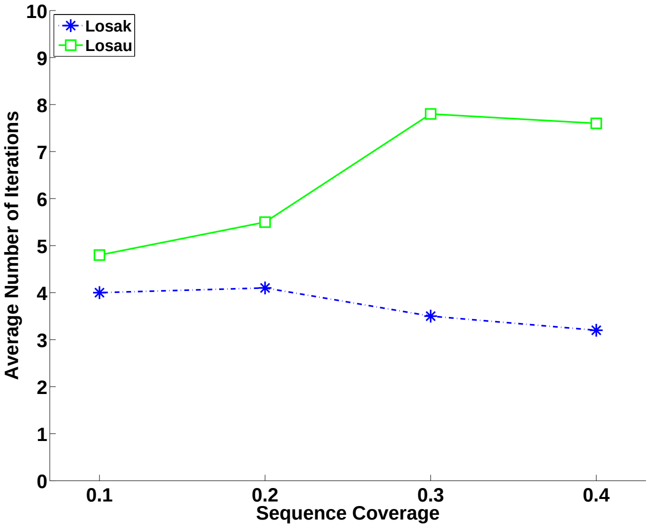

Fig. 6. Comparison of Losak and Losau in terms of the number of iterations for the sequence coverage ranging from 0.1 to 0.4.

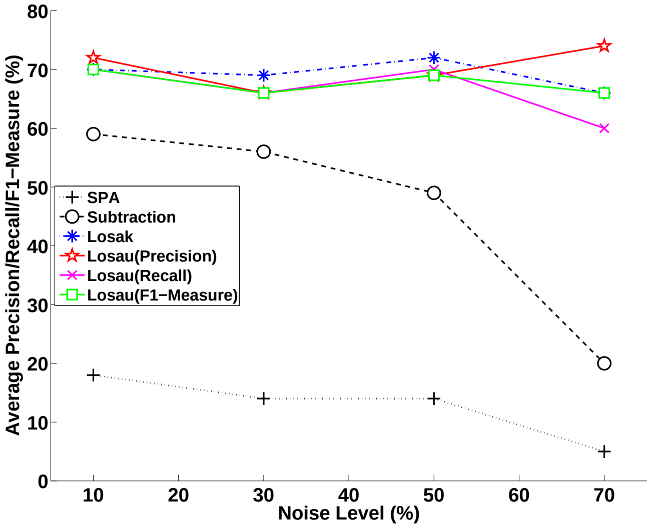

Fig. 7. Identification performance comparison when the ratio of noisy peaks ranges from 10% to 70%.

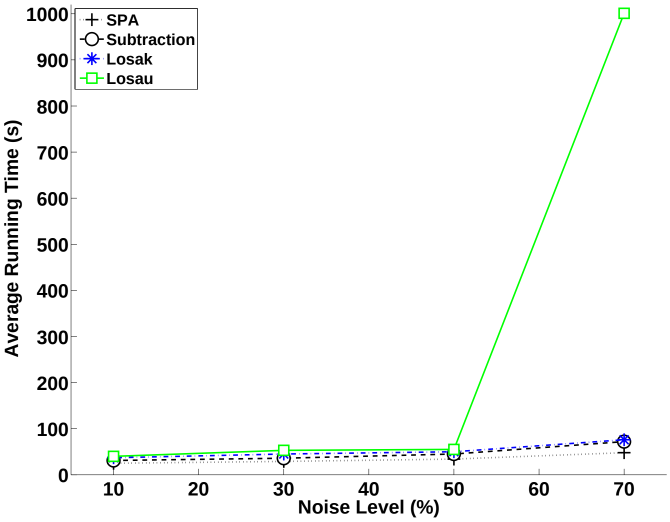

Fig. 8. Running time comparison when the ratio of noisy peaks ranges from 10% to 70%.

at different noise levels. As real MS data can be more complicated than the simulation data, it is not surprising that the Subtraction algorithm will fail.

Fig.8 describes the running time of different methods at different noise levels. All methods take more computational time when the number of noisy peaks increases.

Fig.9 provides a detailed plot of iteration number in both Losak and Losau when the noise level ranges from 10% to 70%. A jump of iteration number in Losau is clearly visible, indicating the extra effort needed for achieving convergence at high noise level.

3.2.5 The Effect of Protein Number in the Mixture

The number of component proteins also has an effect on the performance of PMF algorithms. Generally, the ground-truth peaks of one protein may be considered as noise to other proteins. Therefore, more proteins in the mixture, more difficult the identification.

We set the number of component proteins to 10, 40, 70 and 100, respectively. For each fixed protein number, we generate 10 mixtures using the following parameters: mass error 0.02 Da, sequence coverage 0.3 and no noisy peaks.

Fig.10 depicts that the identification performance of all PMF algorithms declines when the protein number increases. The performance decay of Subtraction algorithm is very fast because it is sensitive to “noise”. Here the “noise” corresponds to peaks digested by other proteins.

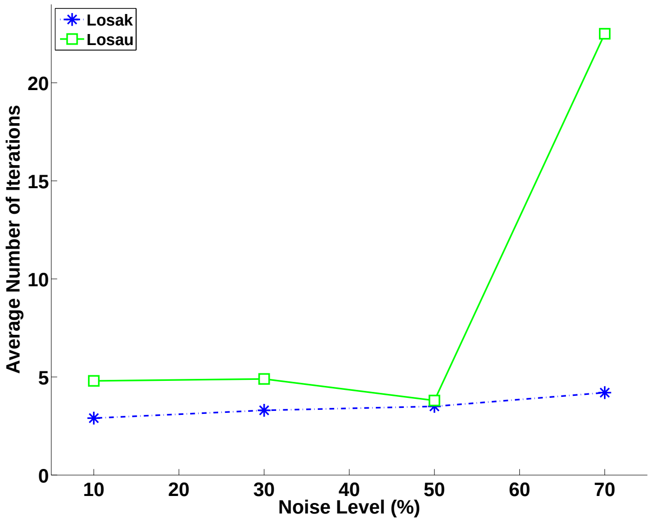

Fig. 9. Comparison of Losak and Losau in terms of the number of iterations when the ratio of noisy peaks ranges from 10% to 70%.

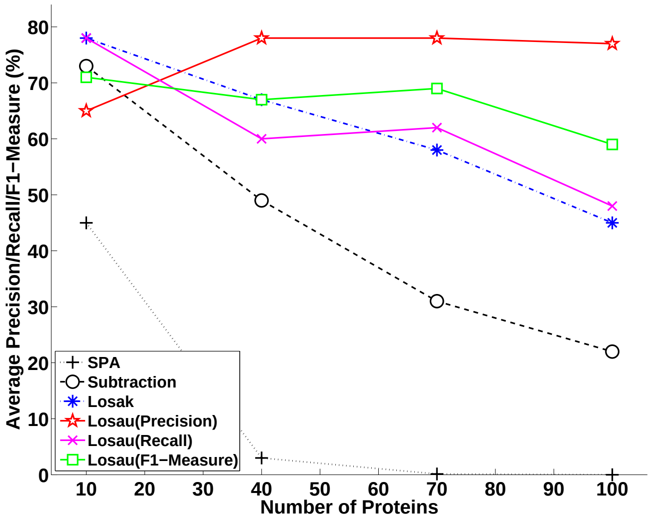

Fig. 10. Identification performance comparison when the number of component proteins in the mixture varies from 10 to 100.

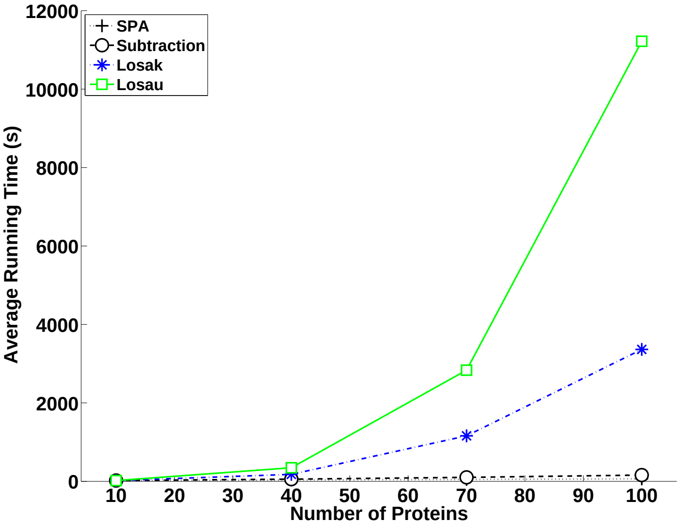

Fig. 11. Running time comparison when the number of component proteins in the mixture varies from 10 to 100.

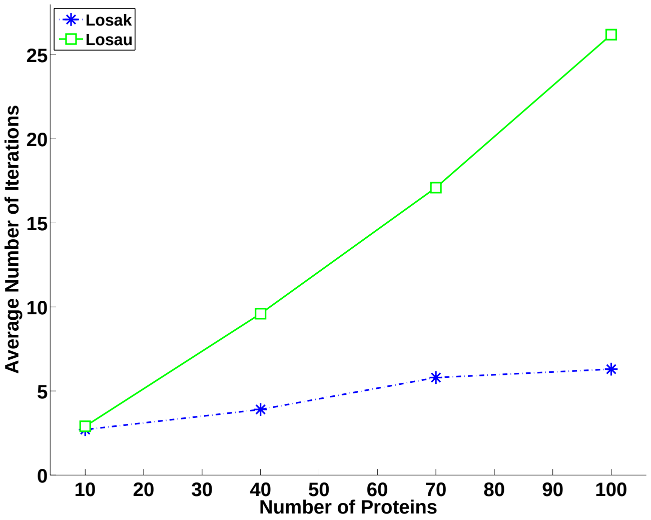

Fig. 12. Comparison of Losak and Losau in terms of the number of iterations when the number of component proteins in the mixture varies from 10 to 100.

Fig.11 shows that the running time of PMF algorithms increases if there are more proteins in the mixture. Note that the computational time of Lsoak and Losau is almost quadratic to the protein number. This is because the increase of protein number will increase number of peaks. In other words, k and l are increased simultaneously so that the running time of both algorithms increases significantly.

We also record the number of iterations used by Lsoak and Losau in Fig. 12. The number of iterations of both algorithms increases linearly and Losau needs more iterations than Losak.

3.3 Real Data

Here we use a mix of 17 proteins generated from a linear ion trap-orbitrap (LTQ-Orbitrap) instrument (Lu *et al.*, 2008). The LTQ-Orbitrap is a hybrid Fourier transform mass spectrometer that combines the efficiency and sensitivity of the linear ion trap with high mass accuracy and high resolution of the orbitrap mass analyzer. After a set of pre-processing steps, we obtain a list of around 6,000 peaks as input. To find more details about sample preparation and data acquisition, the reader is refereed to Lu *et al.* (2008).

In database searching, we set the mass tolerance threshold to 5 ppm according the mass accuracy of the instrument. In Table 1, we report the identification results. Here the number of reported proteins for SPA algorithm, Subtraction algorithm, Losak algorithm is 17, i.e. the number of ground-truth proteins.

Table 1. The performance of different algorithms on the real MS data. Note that all entries for SPA and Subtraction are zeros since they can not identify any ground-truth proteins from the mixture.

Algorithms	*Precision*	*Recall*	*F1-Measure*
SPA	0	0	0
Subtraction	0	0	0
Losak	29%	29%	29%
Losau	9%	29%	14%

Table 1 shows that the optimization-based formulation in Losak and Losau enables us to achieve significant higher protein identification rate in noisy real MS data than previous methods. Certainly, we also notice there is room for further improvement since the best $precision/recall/F1\text{-}measure$ achieved by Losak is approximately 30%. We have the following comments:

(1) The reasons for the unsatisfactory performance are probably two-fold:

- There are too many noisy peaks generated by other components. Recall that the peak list corresponds to a mixture of 17 proteins. Our in-silico digestion indicates that at most 650 peaks will be in the measurement range, yet the input peak list contains around 6000 peaks.
- The sequence coverage is insufficient. As we have observed in the experiment, there are only 13 ground-truth proteins that match more than five peaks in candidate selection process.

Please note that these two reasons are closely related but not identical. In the language of statistics, the first is about false positive, while the second is about true positive.

(2) The data generation process is designed for peptide sequencing method rather than PMF method. It tries to separate peptides of different masses using High-Performance Liquid Chromatography (HPLC) so as to generate MS/MS data effectively. However, such setting creates additional difficulties of combining signals from the same peptide across adjacent scans and insufficient sequence coverage of single protein at each scan.

(3) On the same data, the peptide sequencing method reports more than 800 proteins (Lu *et al.*, 2008). If we evaluate the result in the same way, the $precision$ will be less than 2% although the $recall$ is near 100%.

(4) In addition to those ground-truth proteins, a set of contaminant compounds also exist. If we include these contaminant compounds in the evaluation, the performance of different algorithms will probably increase.

4 DISCUSSIONS

While we have shown that it is possible to achieve an acceptable performance using well-designed PMF algorithms in protein mixtures, it is more important to study how to overcome the bottlenecks hampering the widespread use of PMF method in protein mixture identification. First of all, external factors might be optimized to facilitate successful protein mixture identification. Here we discuss four important factors: mass accuracy, sequence coverage, noise level and protein number in the mixtures.

- Mass accuracy: thanks to the fast development of technology, today's MS instrument such as Fourier transform ion cyclotron resonance mass spectrometer (FT-ICR-MS) has a high mass accuracy capability to measure peptide masses at low ppm levels. This creates the possibility of performing PMF on high-accuracy protein mixture data.
- Sequence coverage: peptide sequencing method faces the same headache of insufficient sequence coverage, while this problem is more serious in the context of PMF. One feasible solution to improve the sequence coverage is to acquire MS data on replicates multiple times so that we are able to detect more peptide signals.
- Noise level: this problem is complicated because noisy peaks always exist. Moreover, these noisy peaks can be generated from various components at different stage. In addition to designing better sample preparation and MS data generation protocol, we need to design more powerful data pre-processing algorithms to reduce false positives in the final peak list.
- Protein number: real-world applications such as biomarker finding need to handle complex MS data, which generally contains thousands of proteins. As shown in our simulation study, the increase of protein number will decrease the performance of PMF algorithms. An obvious solution to this problem is to exploit the separation techniques. Nowadays, HPLC has been frequently used in peptide sequencing method to separate peptide mixtures. In the context of PMF, our objective is to separate proteins instead of peptides. To this end, we need to set up a new experimental protocol and design effective methods to merge the identification results from different separation stages.

On the other hand, it is also possible to obtain better performance through the design of more effective algorithms.

- The local search algorithms in this paper converge to local optimum and provide no performance guarantee. In general, only a small fraction of local optimum are close to the global optimum and the worst local optimum may be of a relatively poor quality. In order to find global optimum, we need to use other meta-heuristics such as simulated annealing and tabu search to jump out local optimum.
- We only consider m/z information in current formulation. The incorporation of additional information such peak intensity and retention time can improve the identification performance as well. Such extensions will lead to more complicated optimization problems and pose challenges for algorithm design.

Overall, we are confident that PMF will become a appealing tool for protein mixture identification.

5 CONCLUSIONS

Through the use of two local search based algorithms, we show that there is a great potential to use PMF as a competing method for protein identification from mixtures. We also discuss the bottlenecks that hamper the widespread use of PMF method for protein mixture identification. Finally, it is promising to overcome these limitations and make PMF a standard tool in protein mixture identification.

ACKNOWLEDGEMENTS

We thank Prof. John R. Yates III and Dr. Bingwen Lu for providing us the LTQ-Orbitrap data. This work was supported with the Competitive Earmarked Research Grant 621707 from the Hong Kong Research Grant Council and a research proposal competition award RPC07/08EG.25 from the Hong Kong University of Science and Technology.

REFERENCES

Aebersold, R., Mann, M. (2003) Mass spectrometry-based proteomics. *Nature*, **422**, 198-207.

Berndt, P., Hobohm, U., Langen, H. (1999) Reliable automatic protein identification from matrix-assisted laser desorption/ionization mass spectrometric peptide fingerprints. *Electrophoresis*, **20**, 3521-3526.

Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K. (1987). Occam's Razor. *Information Processing Letters*, **24**, 377-380.

Clauser, K.R., Baker, P., Burlingame, A.L. (1999) Role of accurate mass measurement(±10 ppm) in protein identification strategies employing MS or MS/MS and database searching. *Analytical Chemistry*, **71**, 2871-2882.

Eriksson, J., Fenyö, D. (2004) Probity: a protein identification algorithm with accurate assignment of the statistical significance of the results. *Journal of Proteome Research*, **3**, 32-36.

Eriksson, J., Fenyö, D. (2005) Protein identification in complex mixtures. *Journal of Proteome Research*, **4**, 387-393.

Eriksson, J., Fenyö, D. (2007) Improving the success rate of proteome analysis by modeling protein-abundance distributions and experimental designs. *Nature Biotechnology*, **25**, 651-655.

Gygi, S.P., Rist, B., Gerber, S.A., Turecek, F., Gelb, M.H., Aebersold, R. (1999) Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nature Biotechnology*, **17**, 994-999.

James, P., Quadroni, M., Carafoli, E., Gonnet, G. (1993) Protein identification by mass profile fingerprinting. *Biochemical and Biophysical Research Communications*, **195**, 58-64.

Jensen, O.N., Podtelejnikov, A.V., Mann, M. (1997) Identification of the components of simple protein mixtures by high-accuracy peptide mass mapping and database searching. *Analytical Chemistry*, **69**, 4741-4750.

Link, A.J., Eng, J., Schieltz, D.M., Carmack, E., Mize, G.J., Morris, D.R., Garvik, B.M., Yates, J.R. (1999) Direct analysis of protein complexes using mass spectrometry. *Nature Biotechnology*, **17**, 676-682.

Lu, B.W., Motoyama, A., Ruse. C., Venable, J., Yates, J.R.III. (2008) Improving protein identification sensitivity by combining MS and MS/MS information for shotgun proteomics using LTQ-Orbitrap high mass accuracy data. *Analytical Chemistry*, **80**, 2018-2025.

Mann, M., Hojrup, P., Roepstorff, P. (1993) Use of mass spectrometric molecular weight information to identify proteins in sequence databases. *Biological Mass Spectrometry*, **22**, 338-345.

Margnin, J., Masselot, A., Menzel, C., Colinge, J. (2004) OLAV-PMF: a novel scoring scheme for high-throughput peptide mass fingerprinting. *Journal of Proteome Research*, **3**, 55-60.

Pappin, D.J., Hojrup, P., Bleasby, A.J. (1993) Rapid identification of proteins by peptide-mass fingerprinting. *Current Biology*, **3**, 327-332.

Park, Z.Y., Russell, D.H. (2001) Identification of individual proteins in complex protein mixtures by high-resolution, high-mass-accuracy MALDI TOF-mass spectrometry analysis of in-solution thermal denaturation/enzymatic digestion. *Analytical Chemistry*, **73**, 2558-2564.

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

Samuelsson, J., Dalevi, D., Levander, F., Rögnvaldsson, T. (2004) Modular, scriptable and automated analysis tools for high-throughput peptide mass fingerprinting. *Bioinformatics*, **20**, 3628-3635.

Siepen, J.A., Keevil, E.J., Knight, D., Hubbard, S.J. (2007) Prediction of missed cleavage sites in tryptic peptides aids protein identification in proteomics. *Journal of Proteome Research*, **6**, 399-408.

Washburn, M.P., Wolters, D., Yates, J.R. (2001) Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nature Biotechnology*, **19**, 242-247.

Wool, A., Smilansky,Z. (2002) Precalibration of matrix-assisted laser desorption/ionization-time of flight spectra for peptide mass fingerprinting. *Proteomics*, **2**, 1365-1373.

Yang, D., Ramkissoon, K., Hamlett, E., Giddings, M.C. (2008) High-accuracy peptide mass fingerprinting using peak intensity data with machine learning. *Journal of Proteome Research*, **7**, 62-69.

Yates, J.R.III., Speicher, S., Griffin, P.R., Hunkapiller, T. (1993) Peptide mass maps: a highly informative approach to protein identification. *Analytical Biochemistry*, **214**, 397-408.

Zhang, W.Z., Chait, B.T. (2000) ProFound: an expert system for protein identification using mass spectrometric peptide mapping information. *Analytical Chemistry*, **72**, 2482-2489.