Protein Inference and Protein Quantification: Two Sides of the Same Coin

Zengyou He

School of Software
Dalian University of Technology

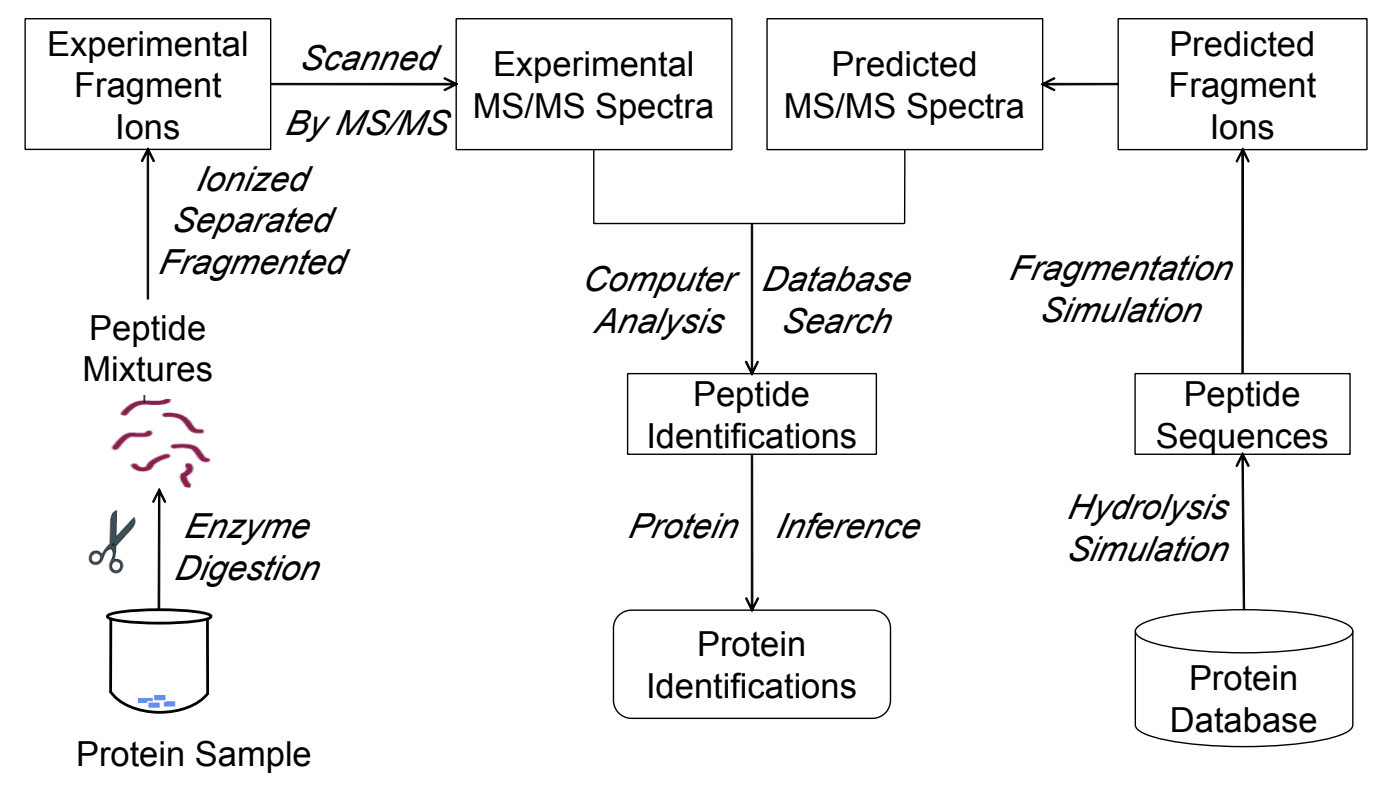
CNCP 2012

Outline

Outline

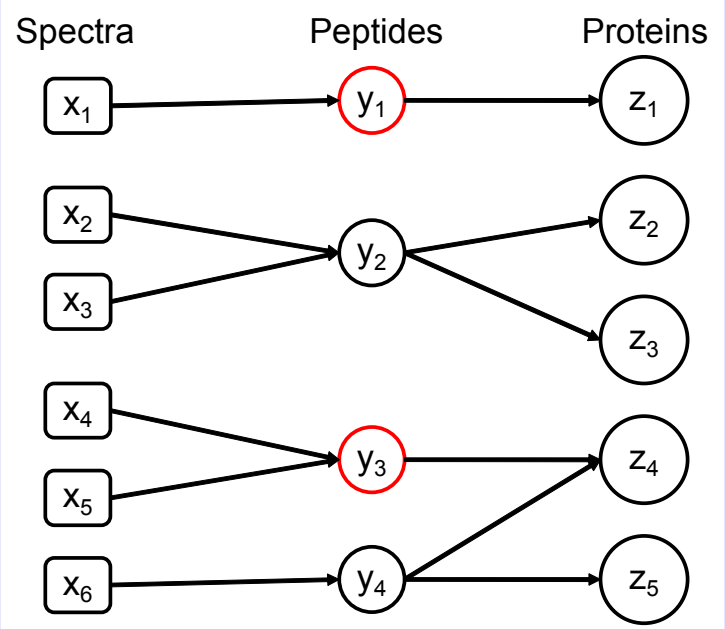
Outline

Protein identification using mass spectrometry in shotgun proteomics



Protein inference

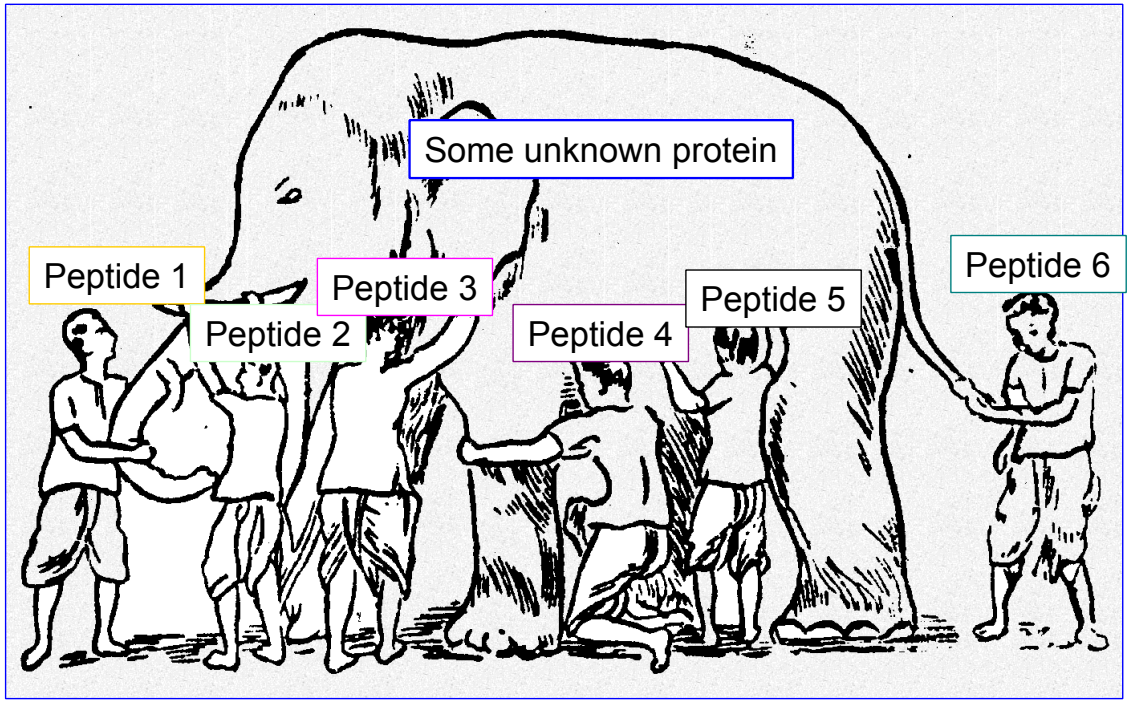
Given peptide identification $(y_1, y_2, \cdots, y_4)$, infer the presence states of the candidate proteins $(z_1, z_2, \cdots, z_5)$.



Why Protein Inference is Important?

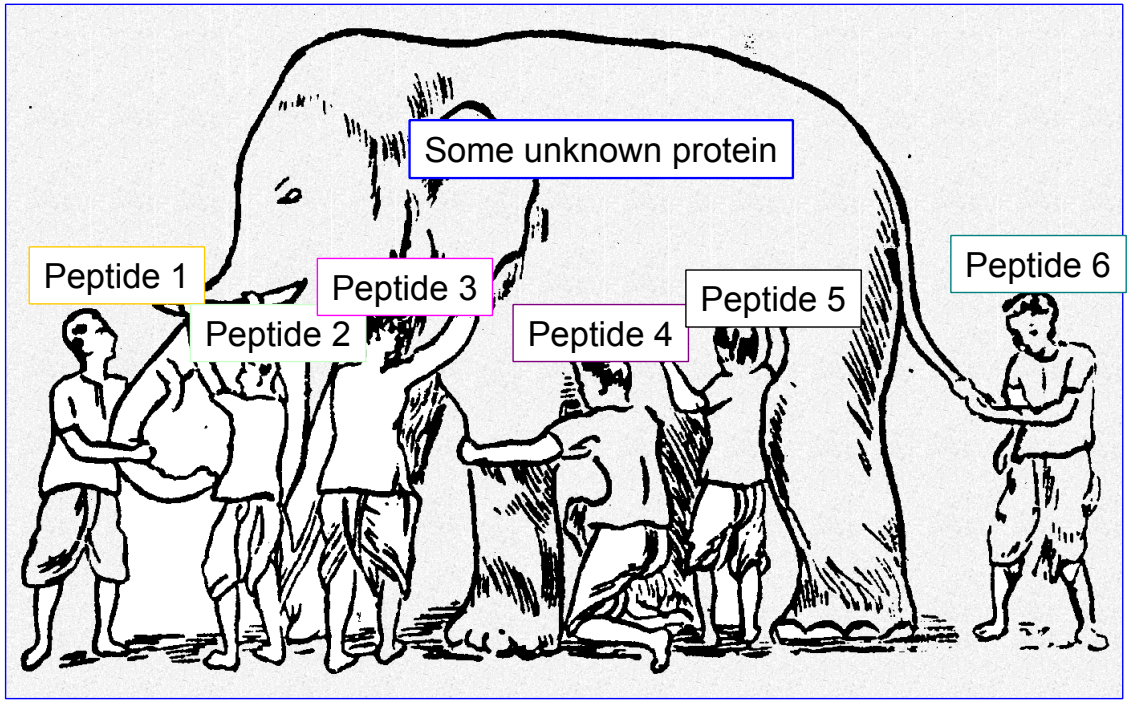
1. Proteins are biologically the most relevant outcome of a shotgun proteomics experiment.
2. The ability of accurately inferring proteins and assessing the inference results is critical to the success of proteomics studies.

Why Protein Inference is Hard?



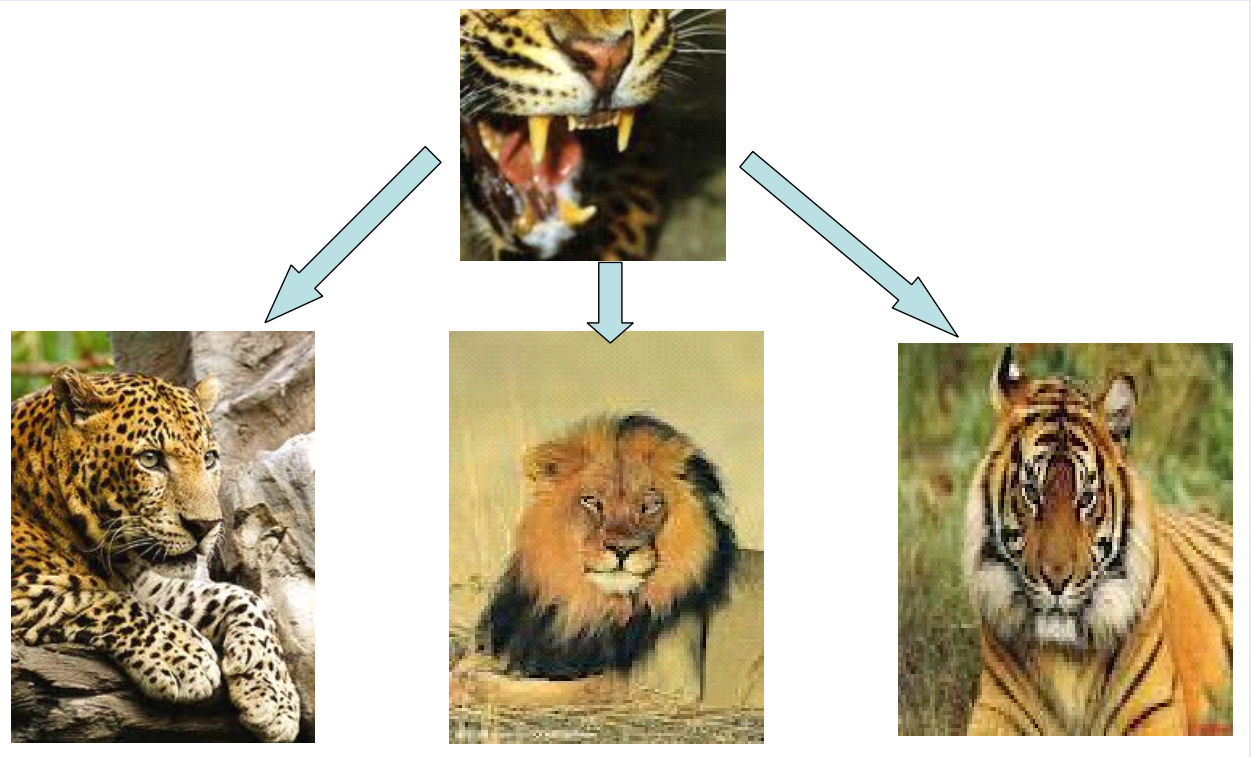
- We have to perform inference with limited information!

Why Protein Inference is Hard?



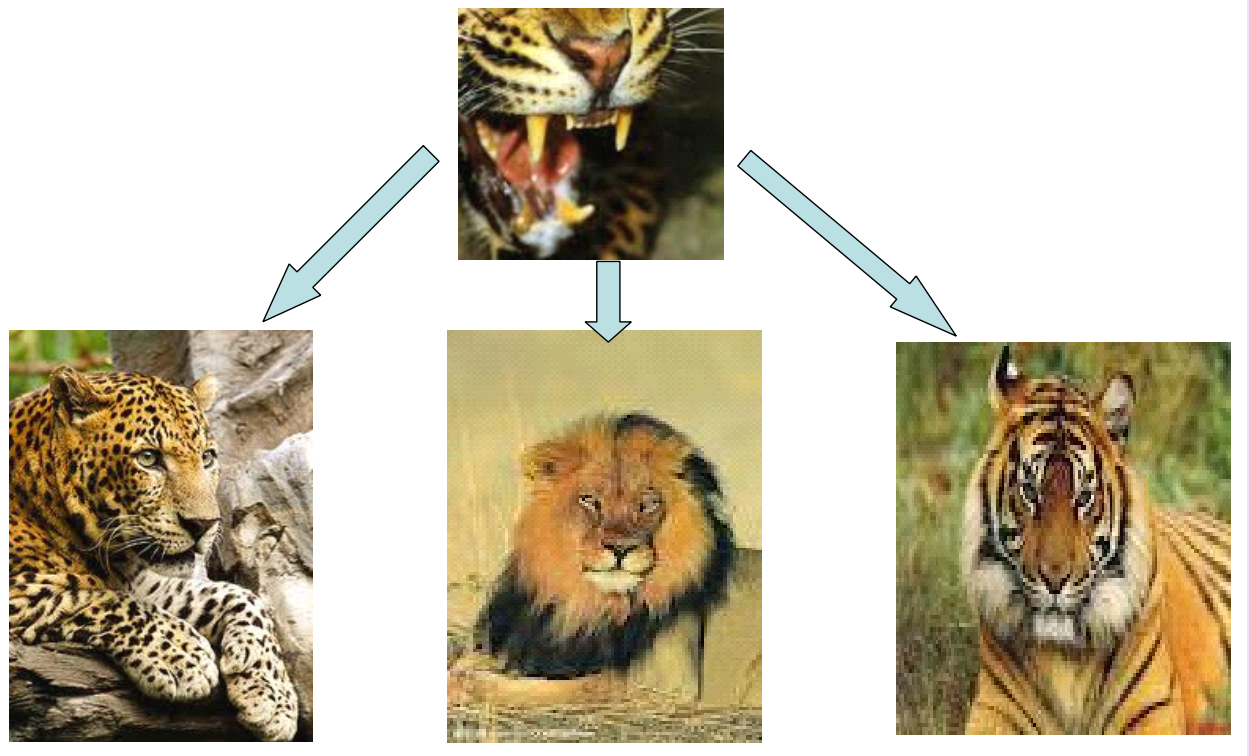
- We have to perform inference with limited information!

Why Protein Inference is Hard?



- We have to perform inference with uncertain information!

Why Protein Inference is Hard?

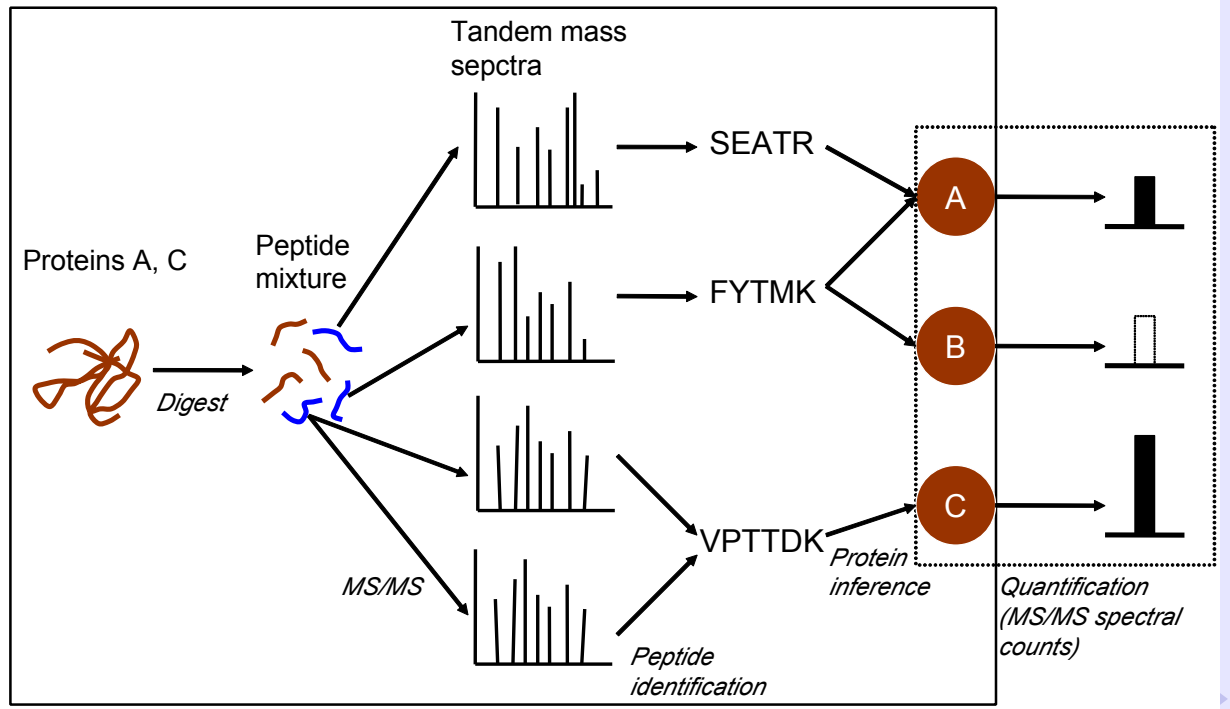


- We have to perform inference with uncertain information!

Outline

Protein Inference and Quantification

Protein identification and quantification have been considered as two individual and subsequent tasks for a long time: first select a subset of proteins that are truly present and then determine the abundances of these proteins.



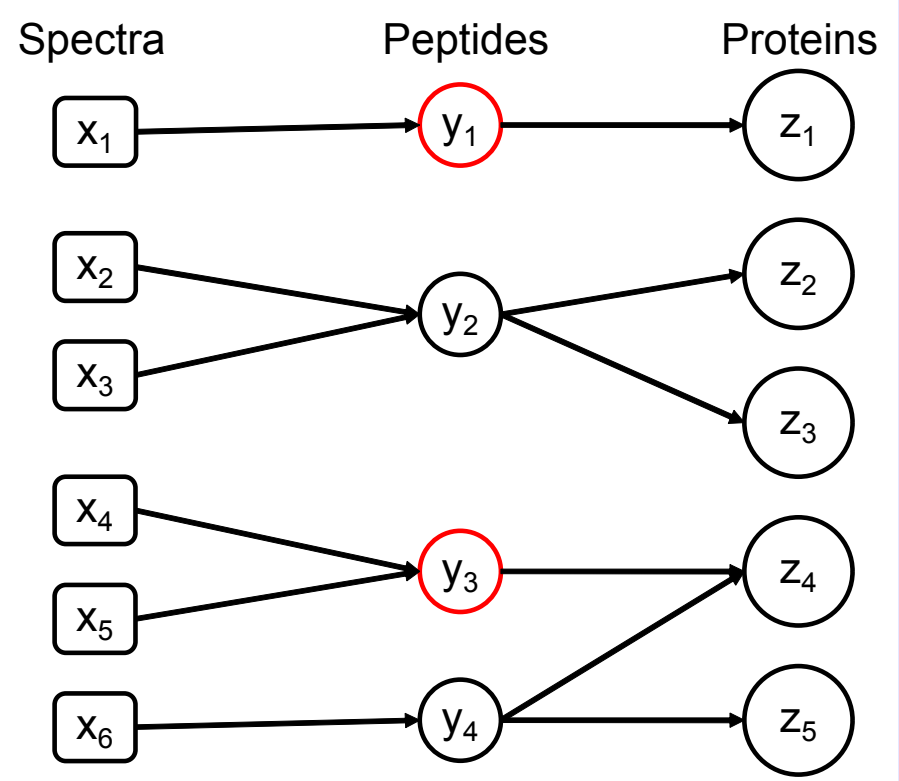
Protein Inference and Quantification

- If one protein is not present, its abundance should be 0. Protein inference problem can be investigated from the perspective of protein quantification: present proteins are those proteins with non-zero abundances.
- We investigate the feasibility of solving protein inference problem with existing protein quantification methods.
- We choose spectral counting as the quantification approach for solving the protein inference problem.

Outline

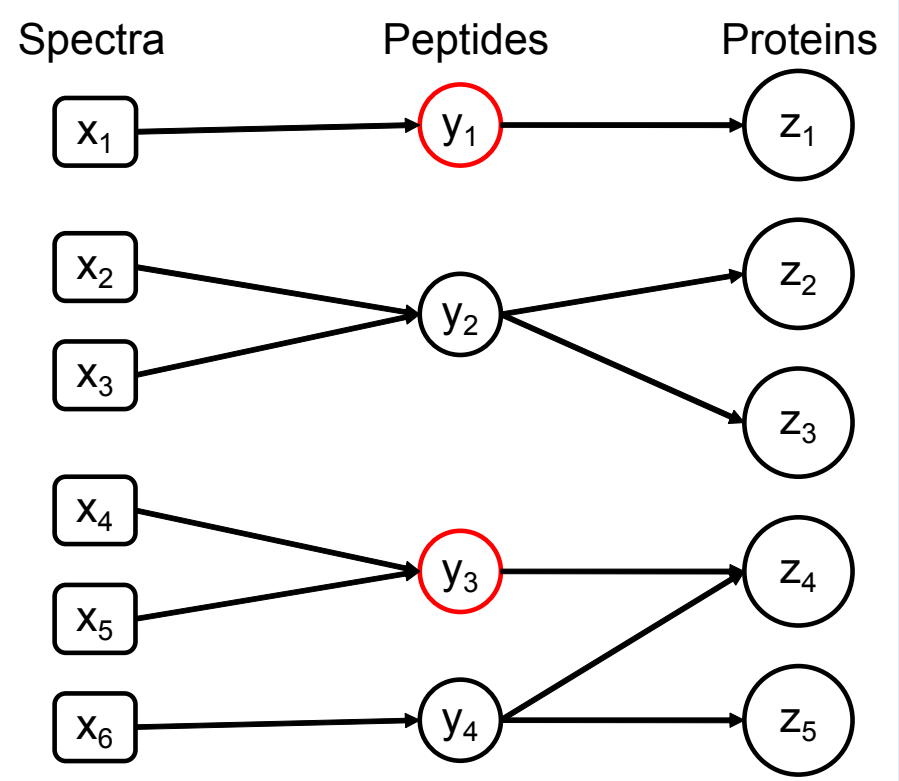
Methods

- The input of the protein inference problem:



Methods

- The input of the protein inference problem:



Methods

1. Multiple Counting: shared peptides are counted multiple times so that the abundances of some proteins may be over-estimated.
2. Equal Division: the abundance of each peptide is distributed equally to different proteins
3. Linear Programming Model: the abundances of some proteins are set to be zero.

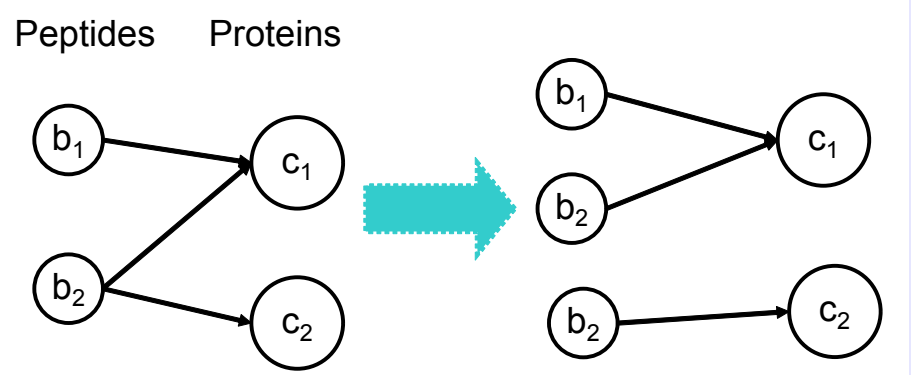
Outline

Multiple Counting

1. The assumption: Shared peptides are used in the same way as the unique peptides and receive no special treatment.
2. The protein abundance is simply the sum of peptide abundance from both shared and unique peptides corresponding to protein z_k:

$$c_k = \sum_{(y_j, z_k) \in E_2} b_j \tag{1}$$

3. $c_1 = b_1 + b_2, c_2 = b_2$

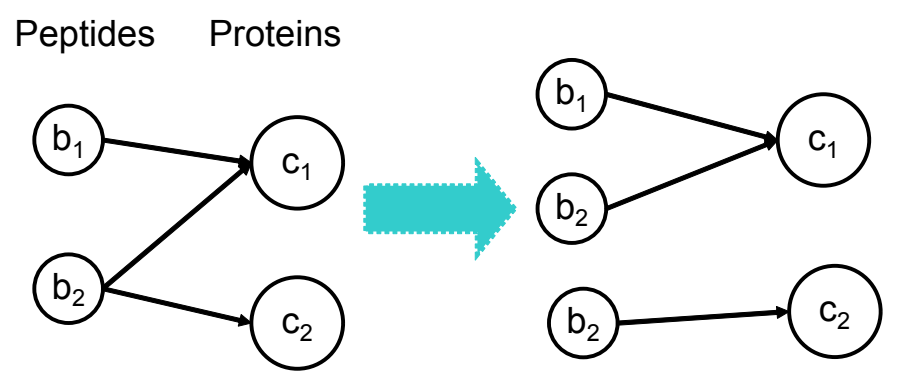


Multiple Counting

1. The assumption: Shared peptides are used in the same way as the unique peptides and receive no special treatment.
2. The protein abundance is simply the sum of peptide abundance from both shared and unique peptides corresponding to protein z_k:

$$c_k = \sum_{(y_j, z_k) \in E_2} b_j \tag{1}$$

3. $c_1 = b_1 + b_2, c_2 = b_2$

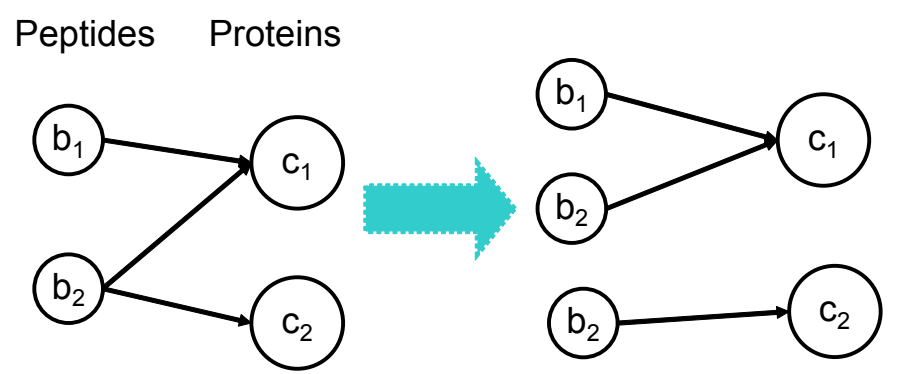


Multiple Counting

1. The assumption: Shared peptides are used in the same way as the unique peptides and receive no special treatment.
2. The protein abundance is simply the sum of peptide abundance from both shared and unique peptides corresponding to protein z_k:

$$c_k = \sum_{(y_j, z_k) \in E_2} b_j \tag{1}$$

3. $c_1 = b_1 + b_2, c_2 = b_2$



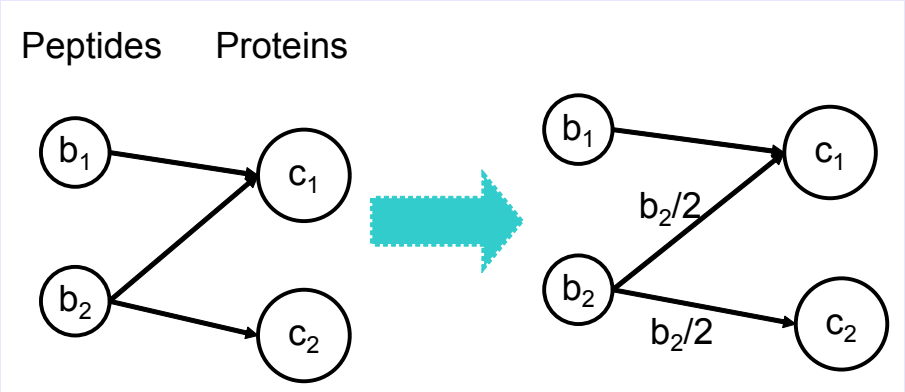
Outline

Equal Division

1. The assumption: Each peptide should be counted only once.
2. The abundance of each shared peptide is equally distributed to its parent proteins:

$$c_k = \sum_{(y_j, z_k) \in E_2} \frac{b_j}{q_j} \tag{2}$$

3. $c_1 = b_1 + \frac{2}{b_2}, c_2 = \frac{2}{b_2}$

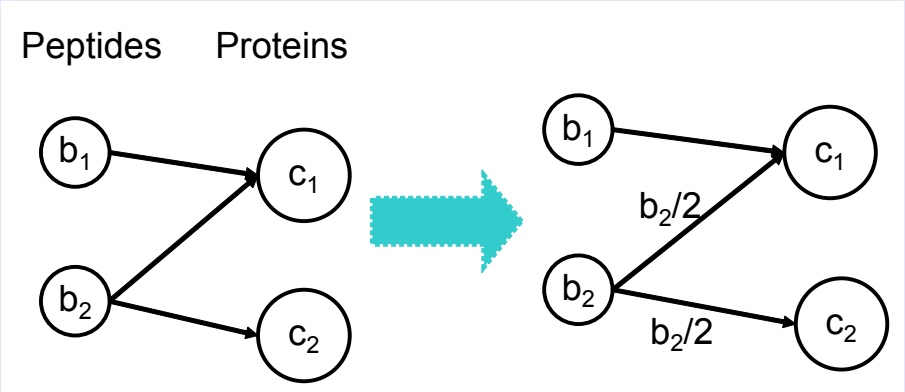


Equal Division

1. The assumption: Each peptide should be counted only once.
2. The abundance of each shared peptide is equally distributed to its parent proteins:

$$c_k = \sum_{(y_j, z_k) \in E_2} \frac{b_j}{q_j} \tag{2}$$

3. $c_1 = b_1 + \frac{2}{b_2}, c_2 = \frac{2}{b_2}$

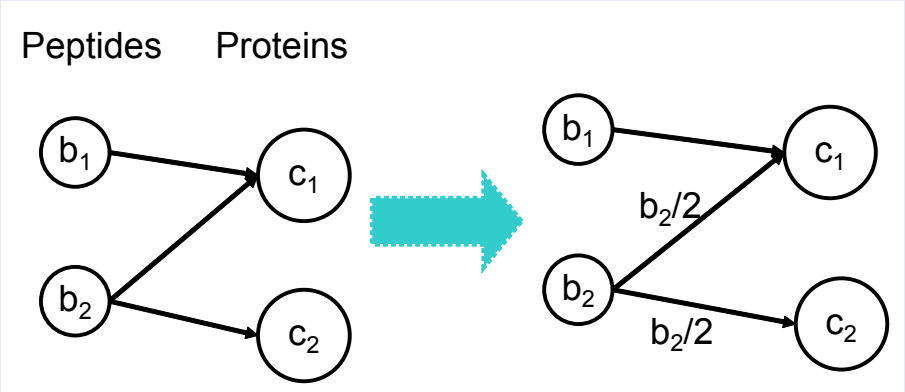


Equal Division

1. The assumption: Each peptide should be counted only once.
2. The abundance of each shared peptide is equally distributed to its parent proteins:

$$c_k = \sum_{(y_j, z_k) \in E_2} \frac{b_j}{q_j} \tag{2}$$

3. $c_1 = b_1 + \frac{2}{b_2}, c_2 = \frac{2}{b_2}$



Outline

Linear Programming Model

1. The assumption: For protein inference problem, some absent proteins should have zero abundances.
2. We first propose a new variable d_{jk} which can be interpreted as the abundance that protein z_k contributes to peptide y_j.
3. For each identified peptide y_j, the peptide abundance can be computed as:

$$b_j = \sum_{\{k|(y_j,z_k)\in E_2\}} d_{jk} \tag{3}$$

Linear Programming Model

1. The assumption: For protein inference problem, some absent proteins should have zero abundances.
2. We first propose a new variable d_{jk} which can be interpreted as the abundance that protein z_k contributes to peptide y_j.
3. For each identified peptide y_j, the peptide abundance can be computed as:

$$b_j = \sum_{\{k|(y_j,z_k)\in E_2\}} d_{jk} \tag{3}$$

Linear Programming Model

1. The assumption: For protein inference problem, some absent proteins should have zero abundances.
2. We first propose a new variable d_{jk} which can be interpreted as the abundance that protein z_k contributes to peptide y_j.
3. For each identified peptide y_j, the peptide abundance can be computed as:

$$b_j = \sum_{\{k|(y_j,z_k)\in E_2\}} d_{jk} \tag{3}$$

Linear Programming Model

We propose a new linear programming model to set the abundances of some proteins to be zero:

$$\min_{D} \sum_{k=1}^{n} t_k \tag{4}$$

$$\forall j, k : d_{jk} \leq t_k \tag{5}$$

$$\forall j : b_j - \sum_{\{k|(y_j,z_k)\in E_2\}} d_{jk} = 0 \tag{6}$$

$$\forall j, k : d_{jk} \sim \begin{cases} = 0 & if\ (y_j, z_k) \notin E_2 \\ \geq 0 & else \end{cases} . \tag{7}$$

Linear Programming Model

Column constraints $\Rightarrow \forall j,k : d_{jk} \le t_k$

$$D=(d_{jk})_{m\times n}=\begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & d_{jk} & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{pmatrix}$$

The variable d_{jk} is interpreted as the abundance that protein z_k contributes to peptide y_j.

Row constraints $\Rightarrow \forall j : b_j - \sum_{\{k|(y_j,z_k)\in E_2\}} d_{jk} = 0$

Linear Programming Model

For each protein z_k, the protein abundance is computed as:

$$c_k = \sum_{\{j|(y_j,z_k)\in E_2\}} d_{jk} \tag{8}$$

Outline

Converting Scores into Probabilities

1. It is beneficial to convert the abundance into well-calibrated probability.
2. The problem of converting ranking scores into estimated probabilities has been widely investigated in different domains.
3. We use the method proposed by Gao et al. [2] to fulfill this task.

Converting Scores into Probabilities

1. It is beneficial to convert the abundance into well-calibrated probability.
2. The problem of converting ranking scores into estimated probabilities has been widely investigated in different domains.
3. We use the method proposed by Gao et al. [2] to fulfill this task.

Converting Scores into Probabilities

1. It is beneficial to convert the abundance into well-calibrated probability.
2. The problem of converting ranking scores into estimated probabilities has been widely investigated in different domains.
3. We use the method proposed by Gao et al. [2] to fulfill this task.

Converting Scores into Probabilities

Given the protein abundance c_k, the probability p_k that protein z_k is present in the sample is estimated as follow:

$$
\begin{aligned}
Pr(z_k = 1|c_k) & \\
&= \frac{Pr(c_k|z_k = 1)Pr(z_k = 1)}{Pr(c_k|z_k = 1)Pr(z_k = 1) + Pr(c_k|z_k = 0)Pr(z_k = 0)} \\
&= \frac{1}{1 + \exp(-f_k)}, \qquad (9)
\end{aligned}
$$

Where

$$
f_k = \log \frac{Pr(c_k|z_k = 1)Pr(z_k = 1)}{Pr(c_k|z_k = 0)Pr(z_k = 0)}. \qquad (10)
$$

Converting Scores into Probabilities

Assuming f_k has a Gaussian distribution with equal covariance matrices, the equation to estimate p_k becomes

$$p_k = \frac{1}{1 + \exp(Ac_k + B)} \tag{11}$$

- Our task becomes to learn the parameters, A and B!

Learning A and B

1. $R = (r_1, r_2, \cdots, r_n)$ is the presence indicator vector of n candidate proteins. Let $r_k = 1$ if protein z_k is present in the sample and 0 otherwise.
2. Under the assumption that the existence of each protein is independent with other proteins, the probability of observing R given $C = \{c_1, c_2, \cdots, c_n\}$ is:

$$Pr(R|C) = \sum_{k=1}^{n} p_k^{r_k}(1 - p_k)^{1-r_k} \tag{12}$$

3. The optimal parameter values should minimize the following negative log likelihood function:

$$LL(R|C) = \sum_{k=1}^{n} [(1 - r_k)(-Ac_k - B) + \log(1 + \exp(Ac_k + B))]$$

Learning A and B

1. $R = (r_1, r_2, \cdots, r_n)$ is the presence indicator vector of n candidate proteins. Let $r_k = 1$ if protein z_k is present in the sample and 0 otherwise.
2. Under the assumption that the existence of each protein is independent with other proteins, the probability of observing R given $C = \{c_1, c_2, \cdots, c_n\}$ is:

$$Pr(R|C) = \sum_{k=1}^{n} p_k^{r_k}(1 - p_k)^{1-r_k} \tag{12}$$

3. The optimal parameter values should minimize the following negative log likelihood function:

$$LL(R|C) = \sum_{k=1}^{n} [(1 - r_k)(-Ac_k - B) + \log(1 + \exp(Ac_k + B))]$$

Learning A and B

1. $R = (r_1, r_2, \cdots, r_n)$ is the presence indicator vector of n candidate proteins. Let $r_k = 1$ if protein z_k is present in the sample and 0 otherwise.
2. Under the assumption that the existence of each protein is independent with other proteins, the probability of observing R given $C = \{c_1, c_2, \cdots, c_n\}$ is:

$$Pr(R|C) = \sum_{k=1}^{n} p_k^{r_k}(1 - p_k)^{1-r_k} \tag{12}$$

3. The optimal parameter values should minimize the following negative log likelihood function:

$$LL(R|C) = \sum_{k=1}^{n} [(1 - r_k)(-Ac_k - B) + \log(1 + \exp(Ac_k + B))]$$

EM algorithm

1. In protein inference problem, the indicator vector R is unknown. Thus, r_k is considered as hidden variables and we employ an EM algorithm to simultaneously estimate A, B and R.
2. The EM algorithm utilizes an iterative procedure to estimate the parameter values $\theta = \{A, B\}$.
3. The procedure includes two steps: set $r_k^{s+1} = E(r_k^s | C, \theta^s)$ (E-step) and compute $\theta^{s+1} = \arg\min_\theta LL(R^{s+1}|C)$ (M-step) where s is the iteration index.

EM algorithm

1. In protein inference problem, the indicator vector R is unknown. Thus, r_k is considered as hidden variables and we employ an EM algorithm to simultaneously estimate A, B and R.
2. The EM algorithm utilizes an iterative procedure to estimate the parameter values $\theta = \{A, B\}$.
3. The procedure includes two steps: set $r_k^{s+1} = E(r_k^s | C, \theta^s)$ (E-step) and compute $\theta^{s+1} = \arg\min_\theta LL(R^{s+1}|C)$ (M-step) where s is the iteration index.

EM algorithm

1. In protein inference problem, the indicator vector R is unknown. Thus, r_k is considered as hidden variables and we employ an EM algorithm to simultaneously estimate A, B and R.
2. The EM algorithm utilizes an iterative procedure to estimate the parameter values $\theta = \{A, B\}$.
3. The procedure includes two steps: set $r_k^{s+1} = E(r_k^s | C, \theta^s)$ (E-step) and compute $\theta^{s+1} = \arg\min_\theta LL(R^{s+1}|C)$ (M-step) where s is the iteration index.

EM algorithm

1. E-step: The unknown vector R is replaced by its expected value R^{s+1} under the current estimated parameter values θ^s. $LL(R|C)$ is minimized by setting $r_k = 0$ if $Ac_k + B > 0$ or $r_k = 1$ if $Ac_k + B \leq 0$.
2. M step: Given the R^{s+1} values, a new parameter estimation θ^{s+1} is computed by minimizing $LL(R|C)$. Since $R^s = [r_k^s]$ is fixed, minimizing $LL(R|C)$ with respect to A and B is a two-parameter optimization problem. This kind of problem can be solved using the model-trust algorithm [3].

EM algorithm

1. E-step: The unknown vector R is replaced by its expected value R^{s+1} under the current estimated parameter values θ^s. $LL(R|C)$ is minimized by setting $r_k = 0$ if $Ac_k + B > 0$ or $r_k = 1$ if $Ac_k + B \leq 0$.
2. M step: Given the R^{s+1} values, a new parameter estimation θ^{s+1} is computed by minimizing $LL(R|C)$. Since $R^s = [r_k^s]$ is fixed, minimizing $LL(R|C)$ with respect to A and B is a two-parameter optimization problem. This kind of problem can be solved using the model-trust algorithm [3].

Outline

Experimental Results

1. 6 data sets:
   - 3 data sets with known reference sets: Mixture of 18 Purified Proteins; Sigma49; Yeast.
   - 3 data sets without reference sets: D. melanogaster Dataset (DME); HumanMD; HumanEKC.
2. 5 experimental methods:
   - Our methods: multiple counting (MP); equal division (ED); linear programming (LP)
   - Compared methods: MSRaveler 6 (MS6); Yeast Analyzer (YA)

Experimental Results

1. 6 data sets:
   - 3 data sets with known reference sets: Mixture of 18 Purified Proteins; Sigma49; Yeast.
   - 3 data sets without reference sets: D. melanogaster Dataset (DME); HumanMD; HumanEKC.
2. 5 experimental methods:
   - Our methods: multiple counting (MP); equal division (ED); linear programming (LP).
   - Compared methods: MSBayesPro (MSB); YeroenKopher (PP).

Experimental Results

1. 6 data sets:
   - 3 data sets with known reference sets: Mixture of 18 Purified Proteins; Sigma49; Yeast.
   - 3 data sets without reference sets: D. melanogaster Dataset (DME); HumanMD; HumanEKC.
2. 5 experimental methods:
   - Our methods: multiple counting (MP); equal division (ED); linear programming (LP).
   - Compared methods: MSBayesPro (MSB); ProteinProphet (PP).

Experimental Results

1. 6 data sets:
   - 3 data sets with known reference sets: Mixture of 18 Purified Proteins; Sigma49; Yeast.
   - 3 data sets without reference sets: D. melanogaster Dataset (DME); HumanMD; HumanEKC.
2. 5 experimental methods:
   - Our methods: multiple counting (MP); equal division (ED); linear programming (LP).
   - Compared methods: MSBayesPro (MSB); ProteinProphet (PP).

Experimental Results

1. 6 data sets:
   - 3 data sets with known reference sets: Mixture of 18 Purified Proteins; Sigma49; Yeast.
   - 3 data sets without reference sets: D. melanogaster Dataset (DME); HumanMD; HumanEKC.
2. 5 experimental methods:
   - Our methods: multiple counting (MP); equal division (ED); linear programming (LP).
   - Compared methods: MSBayesPro (MSB); ProteinProphet (PP).

Experimental Results

1. 6 data sets:
   - 3 data sets with known reference sets: Mixture of 18 Purified Proteins; Sigma49; Yeast.
   - 3 data sets without reference sets: D. melanogaster Dataset (DME); HumanMD; HumanEKC.
2. 5 experimental methods:
   - Our methods: multiple counting (MP); equal division (ED); linear programming (LP).
   - Compared methods: MSBayesPro (MSB); ProteinProphet (PP).

Identification performance comparison (1)

We evaluate the performance using a curve that plots the number of TPs as a function of q-value.

1. An identified protein is labeled as a TP if it is present in the protein reference set or target protein sequence database, and as a FP otherwise.
2. Given a certain probability threshold t, suppose there are T_t TPs and F_t FPs, FDR is estimated as

$$FDR_t = \frac{F_t}{(F_t + T_t)} \tag{14}$$

3. The corresponding q-value is defined as the minimal FDR that a protein is reported:

$$q_t = \min_{t' \leq t} FDR_{t'} \tag{15}$$

Identification performance comparison (1)

We evaluate the performance using a curve that plots the number of TPs as a function of q-value.

1. An identified protein is labeled as a TP if it is present in the protein reference set or target protein sequence database, and as a FP otherwise.
2. Given a certain probability threshold t, suppose there are T_t TPs and F_t FPs, FDR is estimated as

$$FDR_t = \frac{F_t}{(F_t + T_t)} \tag{14}$$

3. The corresponding q-value is defined as the minimal FDR that a protein is reported:

$$q_t = \min_{t' \leq t} FDR_{t'} \tag{15}$$

Identification performance comparison (1)

We evaluate the performance using a curve that plots the number of TPs as a function of q-value.

1. An identified protein is labeled as a TP if it is present in the protein reference set or target protein sequence database, and as a FP otherwise.
2. Given a certain probability threshold t, suppose there are T_t TPs and F_t FPs, FDR is estimated as

$$FDR_t = \frac{F_t}{(F_t + T_t)} \tag{14}$$

3. The corresponding q-value is defined as the minimal FDR that a protein is reported:

$$q_t = \min_{t' \leq t} FDR_{t'} \tag{15}$$

Identification performance comparison (1)

We evaluate the performance using a curve that plots the number of TPs as a function of q-value.

1. An identified protein is labeled as a TP if it is present in the protein reference set or target protein sequence database, and as a FP otherwise.
2. Given a certain probability threshold t, suppose there are T_t TPs and F_t FPs, FDR is estimated as

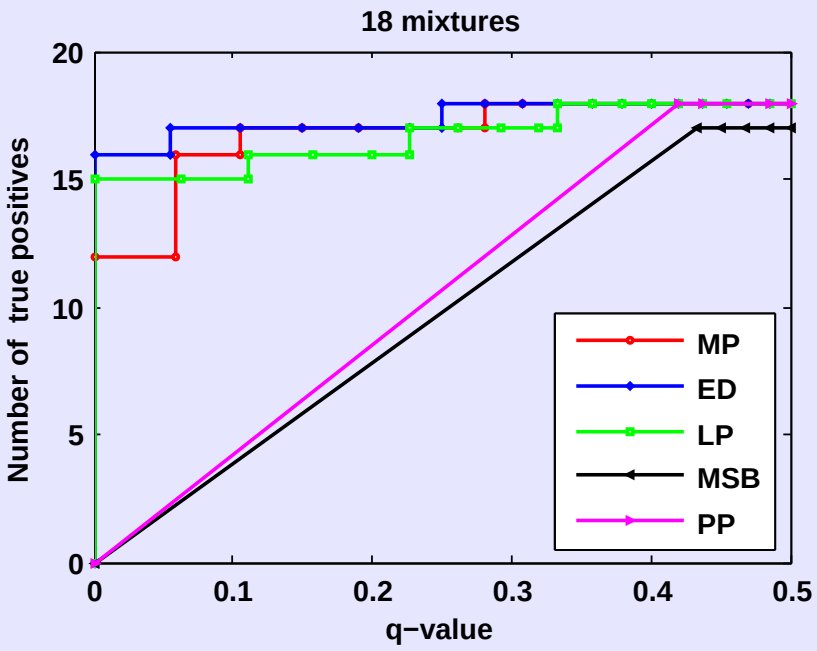
$$FDR_t = \frac{F_t}{(F_t + T_t)} \tag{14}$$

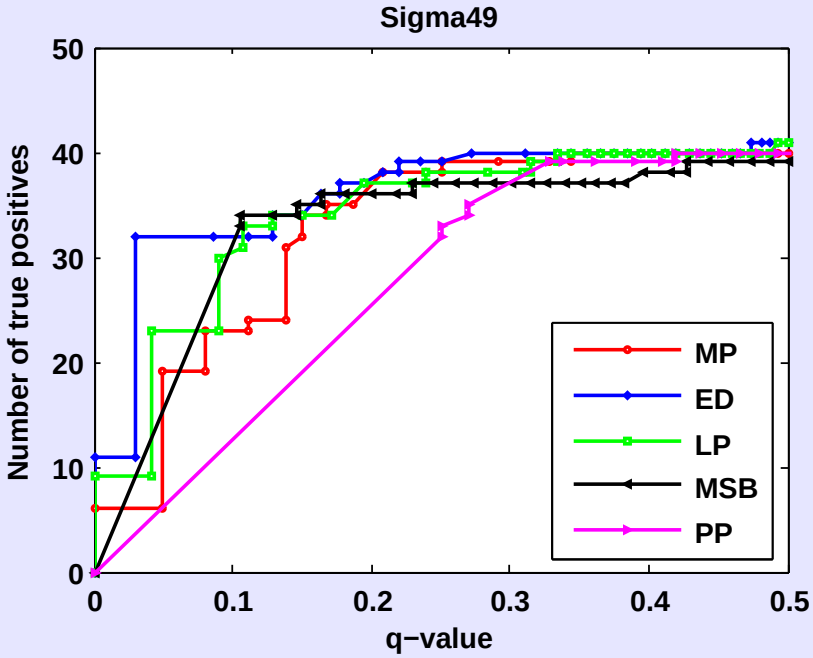
3. The corresponding q-value is defined as the minimal FDR that a protein is reported:

$$q_t = \min_{t' \leq t} FDR_{t'} \tag{15}$$

Identification performance comparison (1)

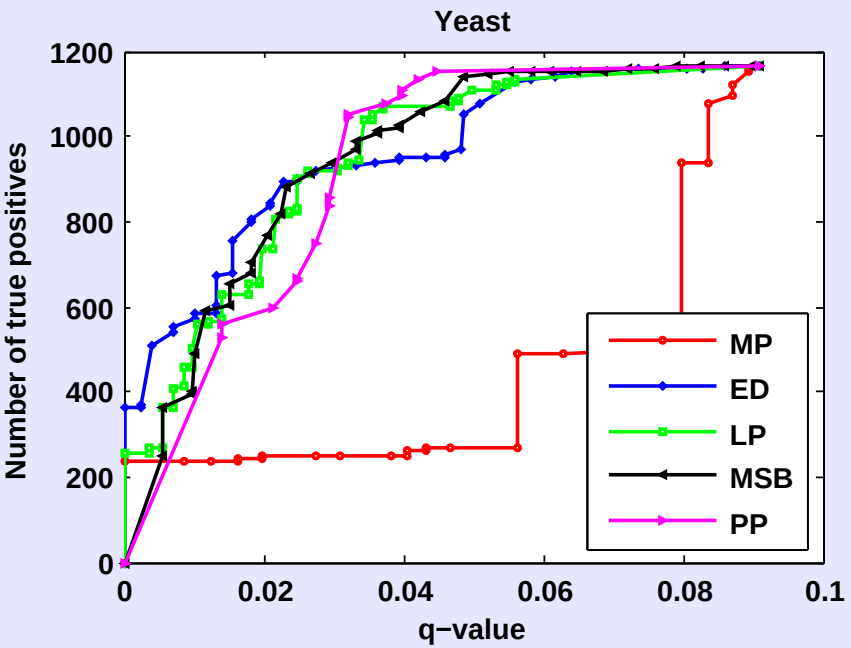
Mixture of 18 Purified Proteins and Sigma49:

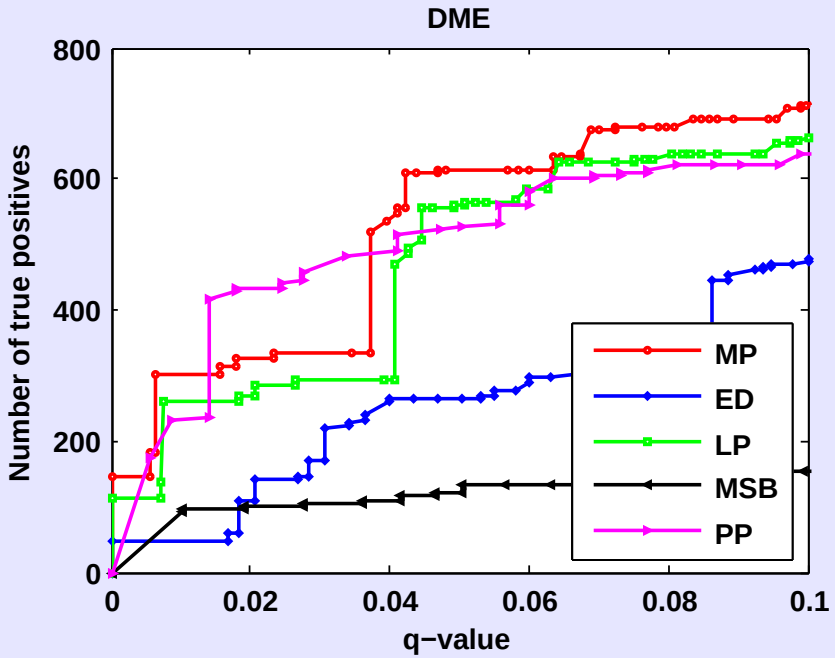




Identification performance comparison (1)

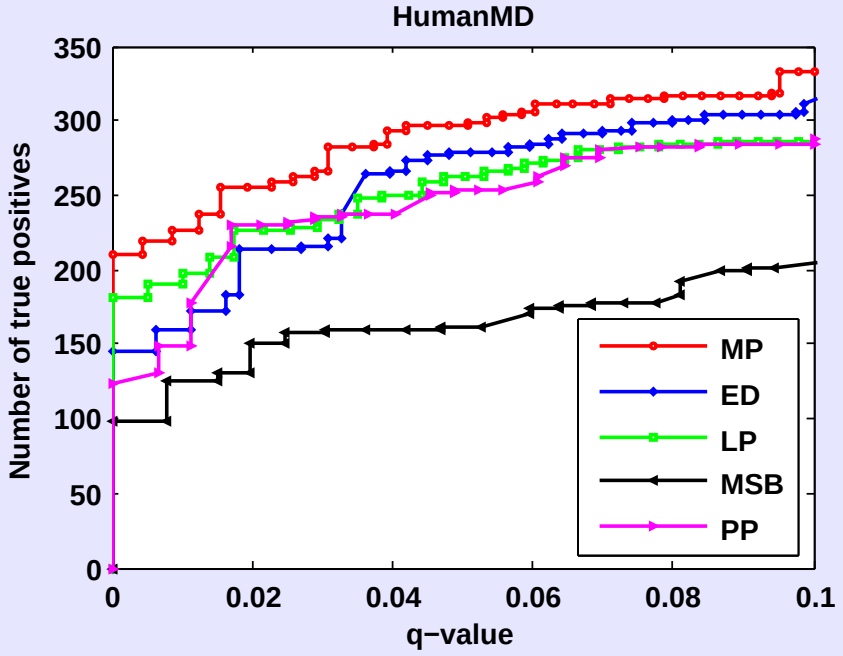
Yeast and DME:

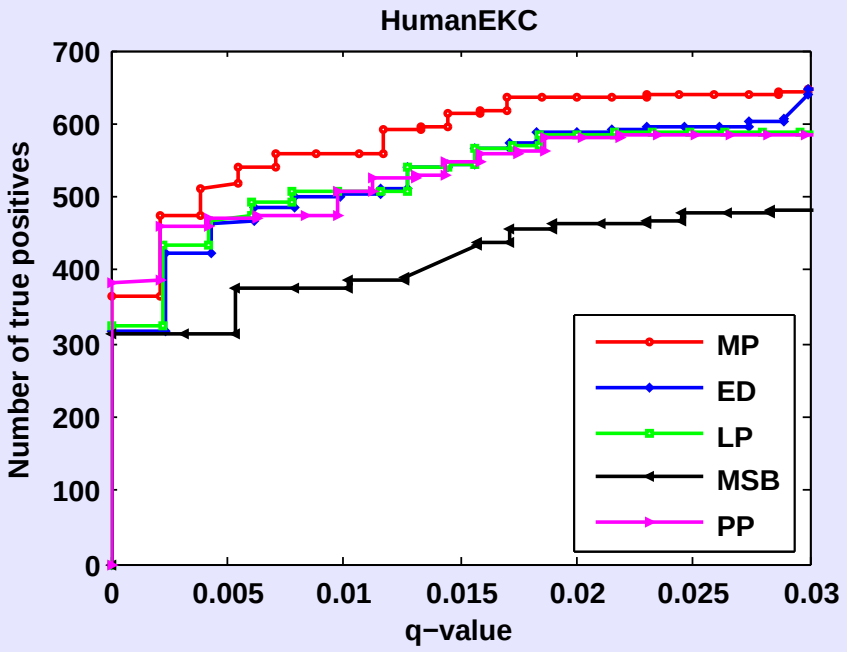




Identification performance comparison (1)

Two human data sets:





Identification performance comparison (2)

- In the calculation of protein abundance, we generalize the number of MS/MS spectra to the sum of PSM probabilities.
- To show the fact of this extension, we compare the identification performance between the generalized spectral counting methods (MP, ED, LP) and the traditional spectral counting methods (NMP, NED, NLP).
- The experimental results indicate that: using the sum of PSM probabilities actually performs better than using the number of PSMs.

Identification performance comparison (2)

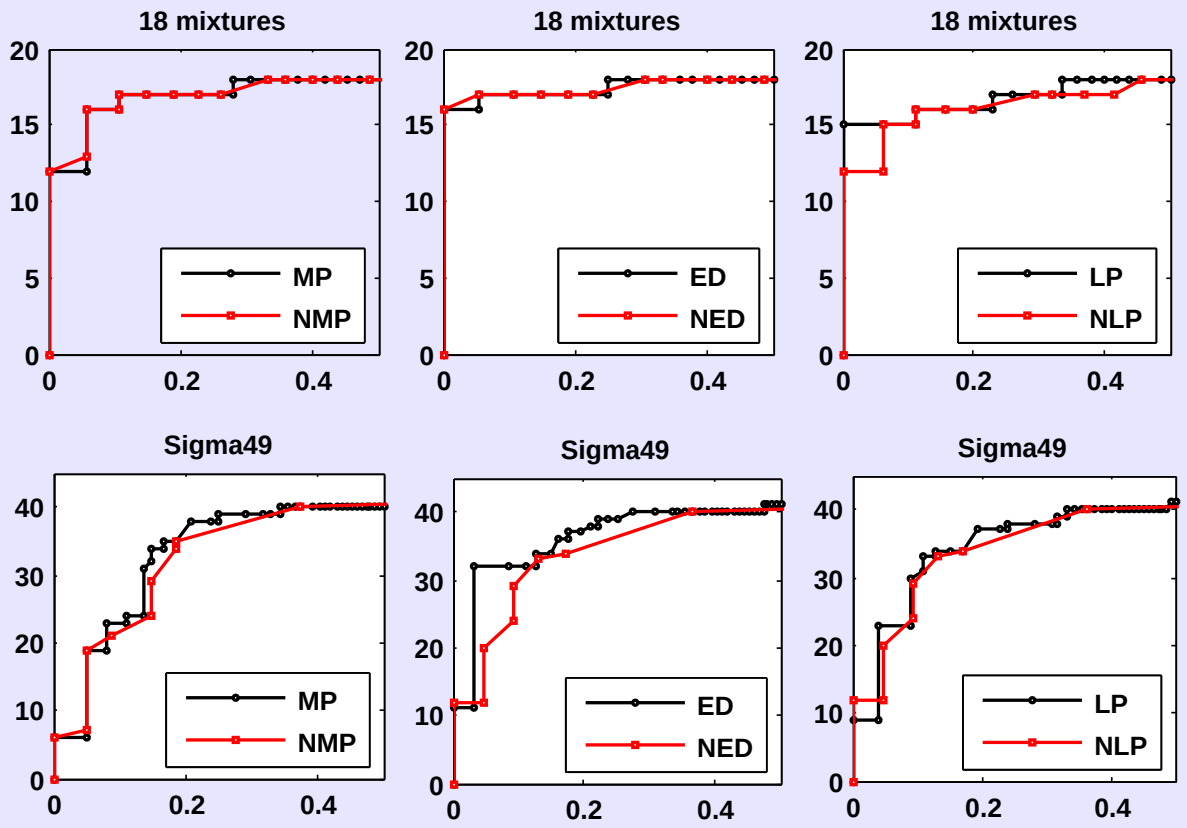
- In the calculation of protein abundance, we generalize the number of MS/MS spectra to the sum of PSM probabilities.
- To show the fact of this extension, we compare the identification performance between the generalized spectral counting methods (MP, ED, LP) and the traditional spectral counting methods (NMP, NED, NLP).
- The experimental results indicate that: using the sum of PSM probabilities actually performs better than using the number of PSMs.

Identification performance comparison (2)

- In the calculation of protein abundance, we generalize the number of MS/MS spectra to the sum of PSM probabilities.
- To show the fact of this extension, we compare the identification performance between the generalized spectral counting methods (MP, ED, LP) and the traditional spectral counting methods (NMP, NED, NLP).
- The experimental results indicate that: using the sum of PSM probabilities actually performs better than using the number of PSMs.

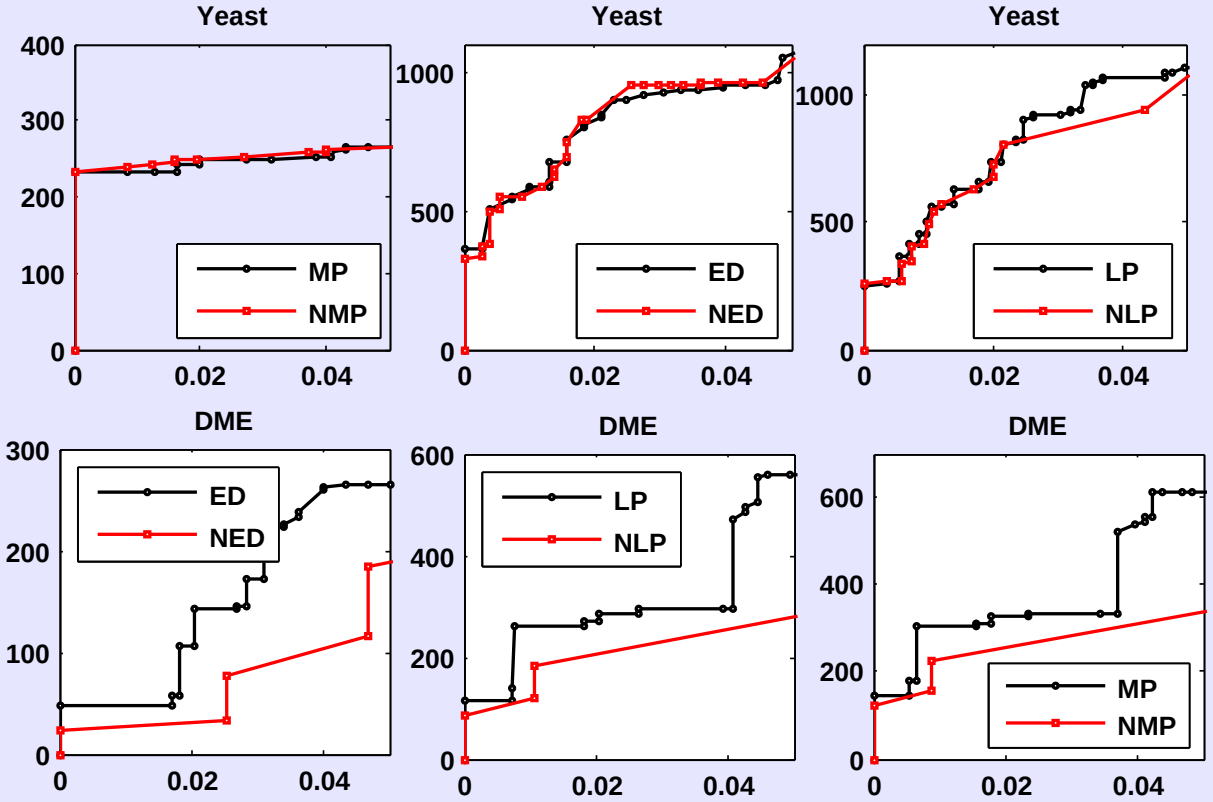
Identification performance comparison (2)

Mixture of 18 Purified Proteins and Sigma49:



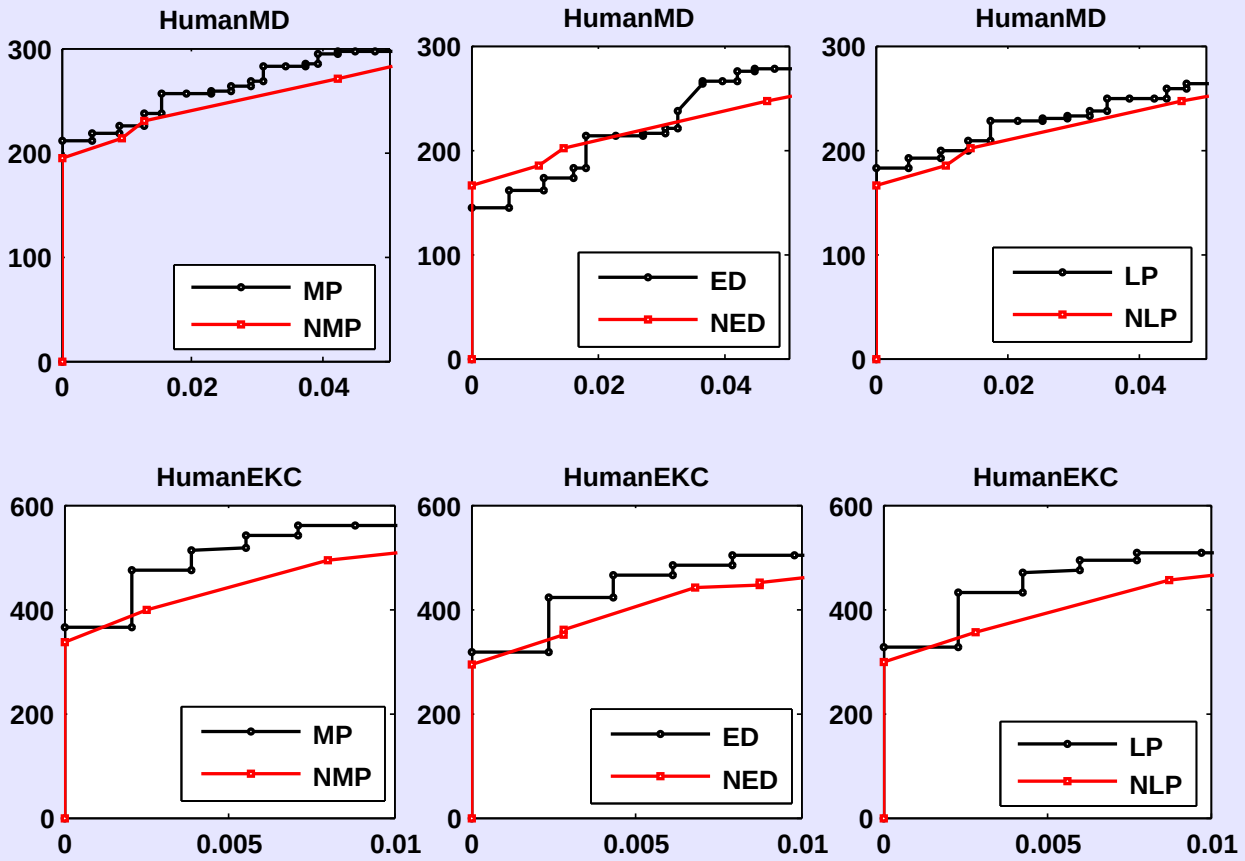
Identification performance comparison (2)

Yeast and DME:



Identification performance comparison (2)

Two human data sets:



Comparison of the score distribution between normalized score and probability estimation

- We use an EM algorithm to convert the abundance score into a well-calibrated probability.
- We compare the distribution of normalized score (NS) and estimated probability (EP).
- The experimental results show that the probability estimation has a more uniform distribution than normalized protein score.

Comparison of the score distribution between normalized score and probability estimation

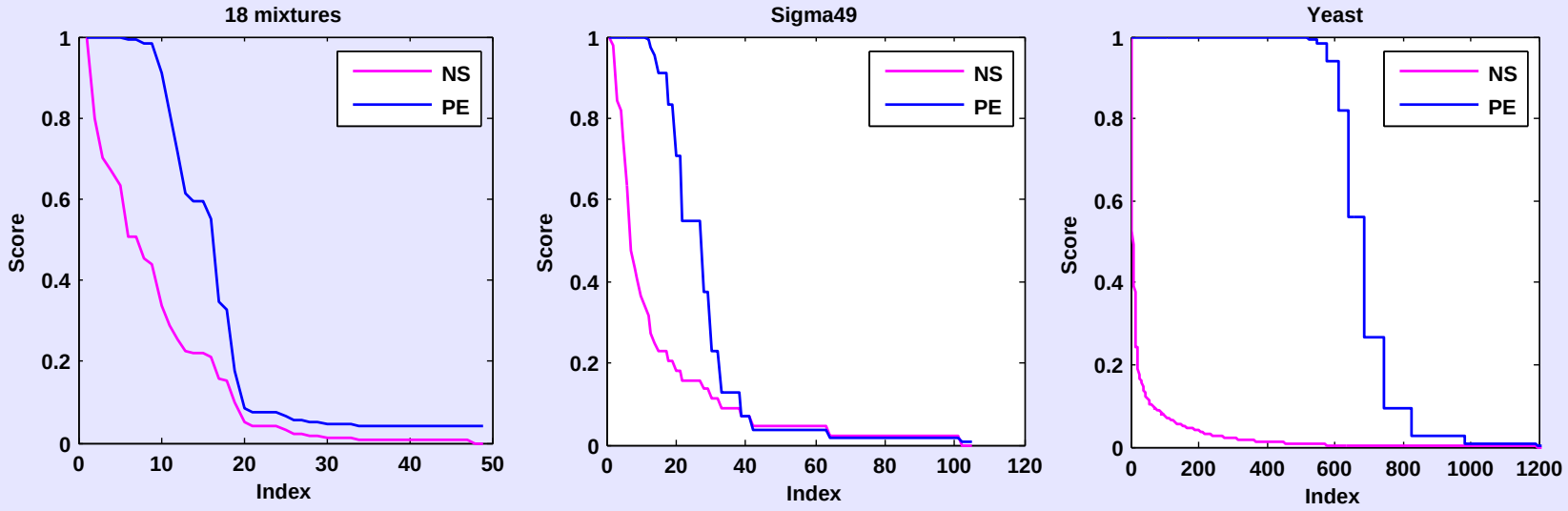
- We use an EM algorithm to convert the abundance score into a well-calibrated probability.
- We compare the distribution of normalized score (NS) and estimated probability (EP).
- The experimental results show that the probability estimation has a more uniform distribution than normalized protein score.

Comparison of the score distribution between normalized score and probability estimation

- We use an EM algorithm to convert the abundance score into a well-calibrated probability.
- We compare the distribution of normalized score (NS) and estimated probability (EP).
- The experimental results show that the probability estimation has a more uniform distribution than normalized protein score.

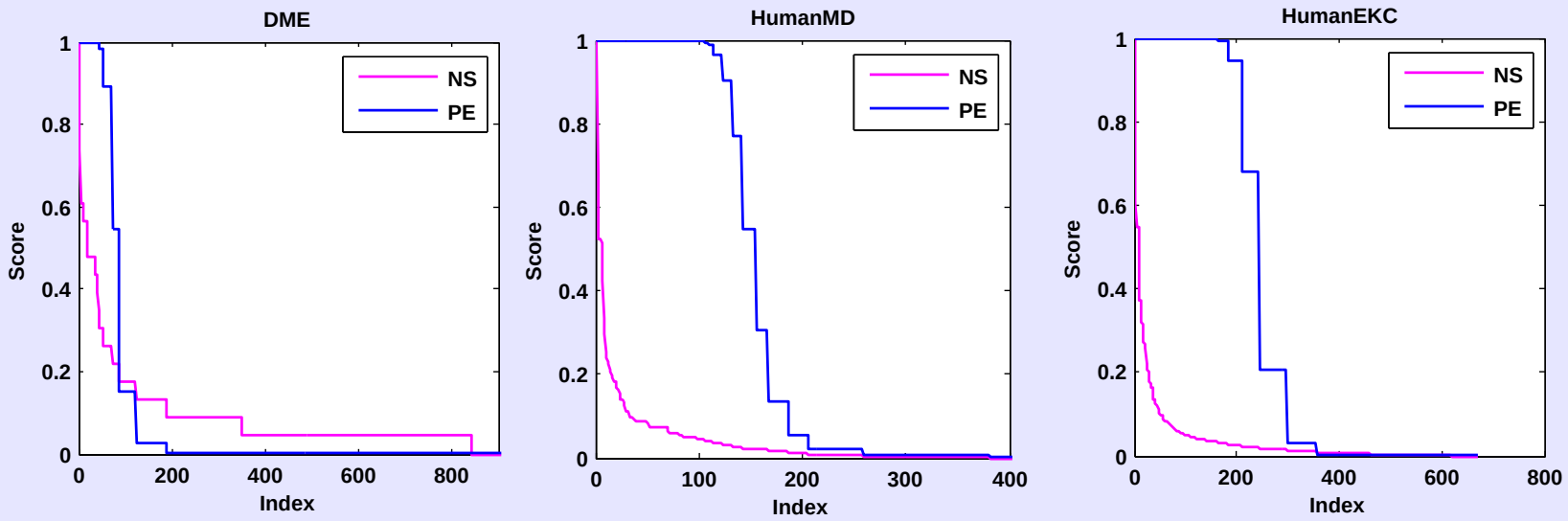
Comparison of the score distribution

Mixture of 18 Purified Proteins, Sigma49 and Yeast



Comparison of the score distribution

DME, HumanMD and HumanEKC



Outline

Conclusion

1. To our knowledge, our method is the first attempt to use protein quantification methods for protein inference.
2. The experimental results show that such a new angle enables us to obtain better identification performance even with some very simple quantification approaches available in the literature.
3. In the future work, we plan to try more quantification methods to check if we can further improve the identification performance.

Conclusion

1. To our knowledge, our method is the first attempt to use protein quantification methods for protein inference.
2. The experimental results show that such a new angle enables us to obtain better identification performance even with some very simple quantification approaches available in the literature.
3. In the future work, we plan to try more quantification methods to check if we can further improve the identification performance.

Conclusion

1. To our knowledge, our method is the first attempt to use protein quantification methods for protein inference.
2. The experimental results show that such a new angle enables us to obtain better identification performance even with some very simple quantification approaches available in the literature.
3. In the future work, we plan to try more quantification methods to check if we can further improve the identification performance.

Reference

- A. I. Nesvizhskii, O. Vitek, and R. Aebersold, "Analysis and validation of proteomic data generated by tandem mass spectrometry," *Nature Methods*, vol. 4, no. 10, pp. 787–797, 2007.
- J. Gao and P.-N. Tan, "Converting output scores from outlier detection algorithms into probability estimates," in *IEEE International Conference on Data Mining*, Hong Kong, China, December 2006, pp. 212–221.
- J. C. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *Advances in Large Margin Classifiers*. MIT Press, 2000, pp. 61–74.