Jinyan Li
Qiang Yang
Ah-Hwee Tan (Eds.)

Data Mining for Biomedical Applications

LNBI 3916

PAKDD 2006 Workshop, BioDM 2006
Singapore, April 2006
Proceedings

Springer

Jinyan Li Qiang Yang Ah-Hwee Tan (Eds.)

Data Mining for Biomedical Applications

PAKDD 2006 Workshop, BioDM 2006
Singapore, April 9, 2006
Proceedings

Springer

Volume Editors

Jinyan Li
Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
E-mail: jinyan@i2r.a-star.edu.sg

Qiang Yang
Hong Kong University of Science and Technology, Department of Computer Science
Clearwater Bay, Kowloon, Hong Kong, China
E-mail: qyang@cs.ust.hk

Ah-Hwee Tan
Nanyang Technological University, School of Computer Engineering
Nanyang Avenue, Singapore 639789
E-mail: asahtan@ntu.edu.sg

Library of Congress Control Number: 2006922190

CR Subject Classification (1998): H.2.8, J.3, I.2, H.3, I.5, F.1

LNCS Sublibrary: SL 8 – Bioinformatics

ISSN	0302-9743
ISBN-10	3-540-33104-2 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-33104-9 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11691730 06/3142 5 4 3 2 1 0

Lecture Notes in Bioinformatics 3916

Edited by S. Istrail, P. Pevzner, and M. Waterman

Subseries of Lecture Notes in Computer Science

Preface

This edited volume contains the papers selected for presentation at the First Workshop on Data Mining for Biomedical Applications (BioDM 2006) held in Singapore on April 9, 2006. The workshop was held in conjunction with the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2006), a leading international conference in the areas of data mining and knowledge discovery. The aim of this workshop was to provide a forum for discussing research topics related to biomedical applications where data mining techniques were found to be necessary and/or useful.

BioDM 2006 received a total of 35 full-length paper submissions from seven countries. Each submitted paper was rigorously reviewed by three Program Committee members. Although many papers were worthy of publication, only 14 regular papers can be accepted in the workshop for presentation and publication in this volume. The accepted papers were organized into three sessions according to their topics, with four papers on database & search, four papers on bio data clustering, and six papers on in-silico diagnosis. The distribution of the paper topics indicated that database query, search, similarity measure, feature selection, and supervised learning remained the current research issues in the field. In addition to the contributed presentation, the BioDM 2006 workshop featured a keynote talk delivered by Limsoon Wong, who shared his insightful vision on the bioinformatics research problems related to protein–protein interactions.

This workshop would not have been possible without the help of many colleagues. We would like to thank the Program Committee members for their invaluable review and comments. Given the extremely tight review schedule, their effort to complete the review reports before the deadline was greatly appreciated. In addition, we found some reviewers' comments were really excellent, as good as what is usually found in a survey paper—critical, constructive, and comprehensive. These comments were very helpful for us in selecting the papers.

Very importantly, we would like to acknowledge the PAKDD 2006 Conference Chair Lim Ee Peng for coordinating with the publisher. Without his effort, these proceedings may not have been published in time for the workshop. We also thank Elaine Koh and Chen Ling for their effort and time in workshop registration and website maintenance.

Thank you all and may the papers collected in the volume inspire your thoughts and research.

April 2006

Jinyan Li
Qiang Yang
Ah-Hwee Tan

Organization

BioDM 2006 workshop was organized by the School of Computer Engineering, Nanyang Technological Univeristy, and Institute for Infocomm Research in conjunction with PAKDD 2006.

Organizing Chair

Ah-Hwee Tan	Nanyang Technological University, Singapore

Program Co-chairs

Jinyan Li	Institute for Infocomm Research, Singapore
Qiang Yang	Hong Kong University of Science and Technology

Program Committee

Keun Ho Ryu	Chungbuk National University, Korea
Kenji Satou	JAIST, Japan
Kenta Nakai	University of Tokyo, Japan
Lusheng Wang	City University of Hong Kong, Hong Kong, China
Qiang Yang	Hong Kong University of Science and Technology, Hong Kong, China
Jingchu Luo	Peking University, China
Juan Liu	Wuhan University, China
Jagath C. Rajapakse	Nanyang Technological University, Singapore
Chee Keong Kwoh	Nanyang Technological University, Singapore
See-Kiong Ng	Institute for Infocomm Research, Singapore
Jinyan Li	Institute for Infocomm Research, Singapore
Vladimir Brusic	University of Queensland, Australia
Yi-Ping Phoebe Chen	Deakin University, Australia
Geoff McLachlan	University of Queensland, Australia
Jian Pei	Simon Fraser Univeristy, Canada
Jianbo Gao	University of Florida, USA
Alexander Statnikov	Vanderbilt University Medical Center, USA
Aik Choon Tan	Johns Hopkins University, USA
Mohammed Zaki	Rensselaer Polytechnic Institute, USA

Lecture Notes in Bioinformatics

Vol. 3916: J. Li, Q. Yang, A.-H. Tan (Eds.), Data Mining for Biomedical Applications. VIII, 155 pages. 2006.

Vol. 3886: E.G. Bremer, J. Hakenberg, E.-H.(S.) Han, D. Berrar, W. Dubitzky (Eds.), Knowledge Discovery in Life Science Literature. XIV, 147 pages. 2006.

Vol. 3745: J.L. Oliveira, V. Maojo, F. Martín-Sánchez, A.S. Pereira (Eds.), Biological and Medical Data Analysis. XII, 422 pages. 2005.

Vol. 3737: C. Priami, E. Merelli, P. Gonzalez, A. Omicini (Eds.), Transactions on Computational Systems Biology III. VII, 169 pages. 2005.

Vol. 3695: M.R. Berthold, R.C. Glen, K. Diederichs, O. Kohlbacher, I. Fischer (Eds.), Computational Life Sciences. XI, 277 pages. 2005.

Vol. 3692: R. Casadio, G. Myers (Eds.), Algorithms in Bioinformatics. X, 436 pages. 2005.

Vol. 3680: C. Priami, A. Zelikovsky (Eds.), Transactions on Computational Systems Biology II. IX, 153 pages. 2005.

Vol. 3678: A. McLysaght, D.H. Huson (Eds.), Comparative Genomics. VIII, 167 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005.

Vol. 3594: J.C. Setubal, S. Verjovski-Almeida (Eds.), Advances in Bioinformatics and Computational Biology. XIV, 258 pages. 2005.

Vol. 3500: S. Miyano, J. Mesirov, S. Kasif, S. Istrail, P.A. Pevzner, M. Waterman (Eds.), Research in Computational Molecular Biology. XVII, 632 pages. 2005.

Vol. 3388: J. Lagergren (Ed.), Comparative Genomics. VII, 133 pages. 2005.

Vol. 3380: C. Priami (Ed.), Transactions on Computational Systems Biology I. IX, 111 pages. 2005.

Vol. 3370: A. Konagaya, K. Satou (Eds.), Grid Computing in Life Science. X, 188 pages. 2005.

Vol. 3318: E. Eskin, C. Workman (Eds.), Regulatory Genomics. VII, 115 pages. 2005.

Vol. 3240: I. Jonassen, J. Kim (Eds.), Algorithms in Bioinformatics. IX, 476 pages. 2004.

Vol. 3082: V. Danos, V. Schachter (Eds.), Computational Methods in Systems Biology. IX, 280 pages. 2005.

Vol. 2994: E. Rahm (Ed.), Data Integration in the Life Sciences. X, 221 pages. 2004.

Vol. 2983: S. Istrail, M.S. Waterman, A. Clark (Eds.), Computational Methods for SNPs and Haplotype Inference. IX, 153 pages. 2004.

Vol. 2812: G. Benson, R.D. M. Page (Eds.), Algorithms in Bioinformatics. X, 528 pages. 2003.

Vol. 2666: C. Guerra, S. Istrail (Eds.), Mathematical Methods for Protein Structure Analysis and Design. XI, 157 pages. 2003.

Table of Contents

Keynote Talk

Database and Search

Bio Data Clustering

In-silico Diagnosis

Exploiting Indirect Neighbours and Topological Weight to Predict Protein Function from Protein-Protein Interactions

Hon Nian Chua, Wing-Kin Sung, and Limsoon Wong

School of Computing and Graduate School for Integrated Sciences and Engineering, National University of Singapore,
3 Science Drive 2, Singapore 117543
{g0306417, dcsswk, dcswls}@nus.edu.sg

Abstract. Most approaches in predicting protein function from protein-protein interaction data utilize the observation that a protein often share functions with proteins that interacts with it (its level-1 neighbours). However, proteins that interact with the same proteins (i.e. level-2 neighbours) may also have a greater likelihood of sharing similar physical or biochemical characteristics. We speculate that two separate forms of functional association accounts for such a phenomenon, and a protein is likely to share functions with its level-1 and/or level-2 neighbours. We are interested to find out how significant is functional association between level-2 neighbours and how they can be exploited for protein function prediction.

We made a statistical study on recent interaction data and observed that functional association between level-2 neighbours is clearly observable. A substantial number of proteins are observed to share functions with level-2 neighbours but not with level-1 neighbours. We develop an algorithm that predicts the functions of a protein in two steps: (1) assign a weight to each of its level-1 and level-2 neighbours by estimating its functional similarity with the protein using the local topology of the interaction network as well as the reliability of experimental sources; (2) scoring each function based on its weighted frequency in these neighbours. Using leave-one-out cross validation, we compare the performance of our method against that of several other existing approaches and show that our method performs well.

J. Li et al. (Eds.): BioDM 2006, LNBI 3916, p. 1, 2006.

A Database Search Algorithm for Identification of Peptides with Multiple Charges Using Tandem Mass Spectrometry

Kang Ning, Ket Fah Chong, and Hon Wai Leong

Department of Computer Science, National University of Singapore,
3 Science Drive 2, Singapore 117543
{ningkang, chongket, leonghw}@comp.nus.edu.sg

Abstract. Peptide sequencing using tandem mass spectrometry is the process of interpreting the peptide sequence from a given mass spectrum. Peptide sequencing is an important but challenging problem in bioinformatics. The advancement in mass spectrometry machines has yielded great amount of high quality spectra data, but the methods to analyze these spectra to get peptide sequences are still accurate. There are two types of peptide sequencing methods –database search methods and the de novo methods. Much progress has been made, but the accuracy and efficiency of these methods are not satisfactory and improvements are urgently needed. In this paper, we will introduce a database search algorithm for sequencing of peptides using tandem mass spectrometry. This Peptide Sequence Pattern (PSP) algorithm first generates the peptide sequence patterns (PSPs) by connecting the strong tags with mass differences. Then a linear time database search process is used to search for candidate peptide sequences by PSPs, and the candidate peptide sequences are then scored by share peaks count. The PSP algorithm is designed for peptide sequencing from spectra with multiple charges, but it is also applicable for singly charged spectra. Experiments have shown that our algorithm can obtain better sequencing results than current database search algorithms for many multiply charged spectra, and comparative results for singly charged spectra against other algorithms.

1 Introduction

As the volume of MS/MS mass spectra grows, the accompanying algorithmic technology for automatically interpreting these spectra has to keep pace. An increasingly urgent problem is the interpretation of multi-charge spectra – MS/MS spectra with charge 3, 4, and 5 are available from the publicly accessible GPM (Global Proteome Machine) dataset [1]. It is foreseen that increasingly there will be more multi-charge spectra produced and so the problem of accurate interpretation of these spectra will become more important with time.

Most existing algorithms for peptide sequencing have been focused largely on interpreting spectra of charge 1. Even when dealing with multiply-charged spectrum, they assume each peak is of charge 1. Only a few algorithms take into account or explicitly make known that they taken into account spectra with charge 2 or higher [2-4].

J. Li et al. (Eds.): BioDM 2006, LNBI 3916, pp. 2 – 13, 2006.

Database searching algorithms [5-8] rely primarily on the completeness of databases, and the availability of a good scoring mechanism. Traditional database search methods have the common principle as this: the experimental spectrum is compared with the theoretical spectrum for each of the peptide in the database, and the peptide from the database with best match usually provides the sequence of the experimental peptide.

The most widely used database search algorithms for analyzing mass spectra of peptides has been software such as SEQUEST [5] and MASCOT [9]. These algorithms search a sequence database for peptides sequences which would produce ions of the mass observed for a particular spectrum, then score these candidate sequences against the observed spectrum. The best match between the peptide tandem mass spectrum and the database-derived peptide sequences is made via a combination of an ion intensity-based score plus a cross-correlation routine. The problems with these algorithms are that they only considered the ions of the mass observed for a particular spectrum, so they can work well for peptide sequences already in the database, but perform badly for peptides with post-translational modifications or other variations.

It is well known that it is almost impossible to find a peptide sequence that *matches exactly* (100% match) with an entry in the database. Instead, many methods rely on matching much shorter sequences called *tags* [6, 10]. However, for some of them [6], the simple assumptions limit the identification accuracy.

In [6], the authors use tag sequence for the search of the peptide sequence. A fragmentation spectrum usually contains a short, easily identifiable series of sequence ions, which yields a partial sequence. This partial sequence divides the peptide into three parts-regions 1, 2, and 3-characterized by the added mass m_1 of region 1, the partial sequence of region 2, and the added mass m_3 of region 3. The construct, m_1 *partial sequence* m_3, is called a "peptide sequence tag" and it is a highly specific identifier of the peptide. An algorithm then uses the sequence tag to find the peptide in a sequence database. The main problem of this approach is that the model used in this algorithm is too simple. A 3-segment peptide sequence tag is used, but not enough to capture several highly-confident fragment sequences. The database search may return several candidates peptide sequences, but further discriminations are very limited.

Because of these problems of the current database search algorithms, it is ideal if we can appropriately utilize all of (or as much as possible) the subsequences (tags) information in the spectrum, and find out the peptide sequence in the database, or detect the post-translational modification that has most support. Recently, the InsPecT algorithm has been developed by Tanner etc. [10], which use more tags information for database search. This algorithm has used score function similar to Dancik score [11] to generate highly reliable tags from spectrum graph, extend tags and use trie to search for candidate peptides in database, and evaluate candidate peptides by statistical analysis. Another database search algorithm based on a set of tags is SPIDER [12].

We have developed a new database algorithm that extend the idea of using tags [6], and we have concentrated on the multi-charge spectrum data. We have tried to utilize all of the tags information, and tried to get the best results based on this information. In our algorithm, we first find out some strong tags from the spectrum, and connect them by their mass differences; these tag-mass combinations are called patterns of the peptide sequence, and the peptide sequences in the database that best match the patterns are selected. This *peptide sequence pattern (PSP)* gives more flexibility and

accuracy to the algorithm, especially for the multiply charged spectra that are very hard to interpret. Then a linear time database search process is used to search candidate peptides sequences by PSPs. These candidate peptide sequences are then scored by share peaks count, ranked and output.

In the following part, we will introduce our formulation of the problem and the database search algorithm. We will then describe our experiment settings and analysis of the results in details.

2 Problem Formulation of Multi-charge Peptide Sequencing

Consider an MS/MS spectrum for a peptide sequence $\rho = (a_1 a_2 ... a_n)$ where a_j is the j^{th} amino acid in the sequence. The *parent mass* of the peptide is given by $m(\rho) = M = \sum_{k=1}^{n} m(a_k)$. Consider a peptide fragment $\rho_k = (a_1 a_2 ... a_k)$, for $k \le n$ that has fragment mass $m(\rho_k) = \sum_{j=1}^{k} m(a_j)$. The peaks in the spectrum come from peptide fragmentation and each peak p can be characterized by the ion-type, specified by $(z, t, h) \in (\Delta_z \times \Delta_t \times \Delta_h)$, where z is the charge of the ion, t is the basic ion-type, and h is the neutral loss incurred by the ion. The set of ion-types considered is $\Delta = (\Delta_z \times \Delta_t \times \Delta_h)$, where $\Delta_z = \{1,2,...,\alpha\}$, $\Delta_t = \{a, b, y\}$, and $\Delta_h = \{\phi, -H_2O, -NH_3\}$. The (z, t, h)-*ion* of the peptide fragment ρ_k will produced an observed peak p in the spectrum S, that has a mass-to-charge ratio of $mz(p)$, that can be computed from the following formula [4]:

$$m(\rho_k) = mz(p) \cdot z + (\delta(t) + \delta(h)) + (z - 1),$$

where $\delta(t)$ and $\delta(h)$ are the mass difference for the respective ion-type and neutral-loss, respectively. The *theoretical spectrum* for ρ is defined in [4] as the set $TS(\rho) = \{p : p$ is observed peak for (z, t, h)-*ion* of peptide fragment ρ_k, for all $(z, t, h) \in \Delta$ and $k=0,1,...,n\}$ of all possible observed peaks for all ion types, and all possible fragments of ρ.

In peptide sequencing, we are given an experimental mass spectrum and the problem is to determine the sequence of the original peptide. A spectrum $S = \{p_1, p_2, \ldots, p_n\}$ of maximum charge α is a set of n peaks where each peak p_k is described by two parameters – $mz(p_k)$, the observed mass-to-charge ratio and $intensity(p_k)$, its intensity. To account for peaks that correspond to ions of charge 2, an "extension" process is performed to convert it to the equivalent peak of charge 1.

The *shared peaks count* between the experimental spectrum S and a peptide ρ is defined as the number of peaks in S that has the same mass as those in $TS(\rho)$, the theoretical spectrum of ρ.

We have followed the computational model in [4]. To account for the different ion-type (especially in multi-charge spectra), [4] introduced the concept of the ***extended spectrum*** S_β^α where α is the maximum charge of the spectrum S, and β is the largest charge considered for extension. In the extended spectrum S_β^α, we "extend" each peak by generating a set of pseudo-peaks (or guesses) that correspond to the different ion-types with charge $\le \beta$. Namely, for each peak $p_j \in S$ and ion-type

$(z,\ t,\ h) \in (\{1,\ldots,\beta\} \times \Delta_t \times \Delta_h)$, we generate pseudo-peak denoted by $(p_j,\ (z,\ t,\ h))$ with a corresponding *assumed* fragment mass given by $m(p_j,(z,t,h)) = mz(p_j)\cdot z + (\delta(t) + \delta(h)) + (z-1)$. A corresponding ***extended spectrum graph*** of connectivity (defined below) d, $G_d(S_\beta^\alpha)$, is also introduced. Each vertex in this graph represents a pseudo-peak $(p_j,\ (z,\ t,\ h))$ in the extended spectrum S_β^α, namely to the $(z,\ t,\ h)$-*ions* for the peak p_j. For simplicity, we also denote the vertex by $(v_j,\ (z,\ t,\ h))$. Each vertex represents a possible peptide fragment mass $m(v_j,\ (z,\ t,\ h))$. An additional notion called the **PRM** (***prefix residue mass***) is also introduced. This mass refers to the prefix mass of the interpreted peptide fragment mass for vertex, and is defined as $PRM_v(v_i)$ = m(v_i) if t(v_i)∈{b-ion} else (a and y-ion) $PRM_v(v_i) = M - \text{m}(v_i)$.

In the "standard" spectrum graph, we have a directed edge $(u,\ v)$ from vertex u to vertex v if $PRM(v)$ is larger than $PRM(u)$ by the mass of a single amino acid. In the extended spectrum graph of connectivity d, $G_d(S_\beta^\alpha)$, we extend the edge connectivity definition to mean "*a directed path of no more than d amino acids*". Thus, we connect vertex u and vertex v by a directed edge $(u,\ v)$ if the $PRM(v)$ is larger than $PRM(u)$ by the total mass of d' amino acids, where $d' \le d$. In this case, we say that the edge (u,v) is connected by a path of length up to d amino edges. Note that the number of possible paths to be searched is 20^d and increased exponentially with d. In practice, we use d=2, except where it is explicitly stated otherwise. We illustrate the extended spectrum graph with an example shown in Fig. 1.

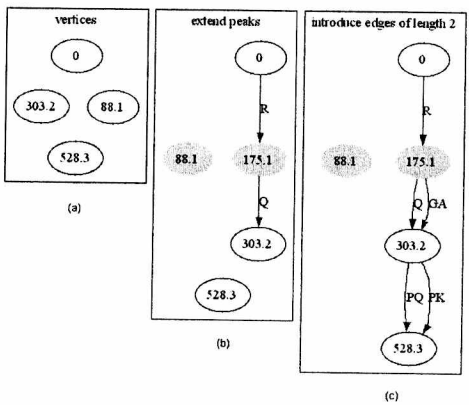

Fig. 1. The difference between $G_1(S_1^2)$ (left) and $G_2(S_2^2)$ (right). There are no paths from v_b to v_e in $G_1(S_1^2)$, but 4 paths in $G_2(S_2^2)$ due to extension.

3 Database Search Algorithm

In our algorithm, we first find out some strong tags from the spectrum, and connect them by their mass differences; these tag-mass combinations are called *Peptide*

Sequence Patterns (PSPs), and the peptide sequences in the database that best match the PSPs are selected for further process. These PSPs give more flexibility and accuracy to the algorithm. Our algorithm is called the PSP algorithm.

Peptide Sequence Patterns Algorithm

The PSP algorithm first compute a set, *BST*, of "best" strong tags. Informally, these strong tags are highly reliable tags found in the spectrum S. To find strong tags, we first restrict the possible ion-types those that appear most frequently. The restricted set of ion-type is given by $\Delta^R = (\Delta_z^R \times \Delta_t^R \times \Delta_h^R)$, where $\Delta_z^R = \{1\}$, $\Delta_t^R = \{b, y\}$, and $\Delta_h^R = \{\phi\}$. Namely, we consider only charge 1, an only *b-ions* and *y-ions* and no neutral loss. We also define $G_1(S_1^\alpha, \{\Delta^R\})$, the extended spectrum graph with ion-type restriction – namely, the spectrum graph $G_1(S_1^\alpha)$ where the ion types considered are restricted to those in Δ^R. Then a *strong tag* T of ion-type $(z, t, h) \in \Delta^R$ is a maximal path $\langle v_0, v_1, v_2, \ldots, v_r \rangle$ in the $G_1(S_1^\alpha, \{\Delta^R\})$, where every vertex $v_i \in T$ is of a (z, t, h)-*ion*. In each component of the graph, PSP algorithm computes a "best" tag with respect to some scoring function. Then the set *BST* is the set comprising the best tag for each component in the spectrum graph. Typically, the number of tags is much smaller than the number of peaks in S. (We refer the reader to [4] for more details.)

Given the set BST of best strong tag, the Peptide Sequence Patterns (PSPs) algorithm then proceeds to find the PSP that result from paths obtained by "connecting" the tags from *BST*. This is done by searching for paths in the graph $G(BST)$ in which the vertices are the strong tags in *BST*, and we have an edge from the tail vertex u of T_1 to the head vertex v of T_2 if the $PRM(v)$ is larger than $PRM(u)$. We note two major difference between $G(BST)$ and the extended spectrum graph – first, the number of vertices in $G(BST)$ is small, and second, the number of edges is also very much smaller since we link strong tags in a head-to-tail manner.

The peptide sequence patterns (PSPs) that represent the paths compose of the tag fragments and mass fragments. Formally, $PSP_i = m_1 t_1 m_2 t_2 \ldots m_n t_n m_{n+1}$, in which m_i and t_i refer to mass difference and tag, respectively. Each tag in the sequence composes of those consecutive amino acids with very high probability to be together. Each mass is the sequence represents the value of masses between tags.

After PSPs are retrieved, the PSPs are scored and ranked according to shared peaks count of the theoretical spectrum of the PSP and the experimental spectrum. Some top PSPs can be selected for database search.

The database search algorithm is essentially an approximation pattern matching in the database, with PSP (composed of tags and mass differences) as pattern. The detailed database search algorithm will be described later.

After database search based on PSPs, several candidate peptides are obtained. For each of candidate peptide sequences, score it by the shared peaks count of the theoretical spectrum of the candidate peptides and the experimental spectrum.

The scheme of the PSP algorithm and the description of the algorithm are illustrated in Fig. 2 and Fig. 3, respectively.

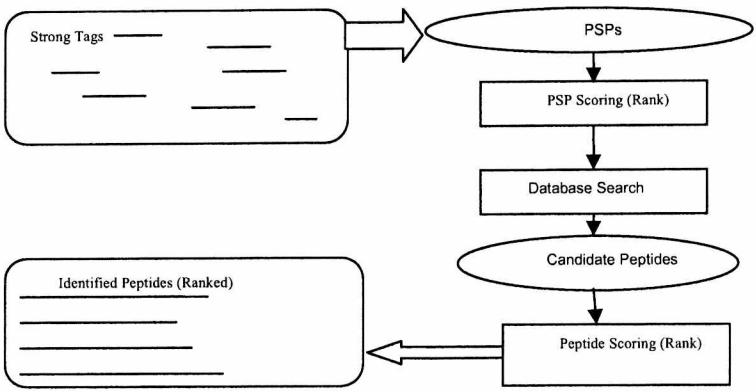

Fig. 2. The scheme of the database search algorithm

1. **Search for strong tags**
   - Transform spectrum to extended spectrum graph
   - Select all of the best strong tags (BST) in extended spectrum graph
2. **Generation of PSPs**
   - Connect BSTs by mass differences
   - Generate a graph G, every vertex is a BST, every edge is one mass difference. Starting and ending vertex represent 0 and parent masses, respectively
   - List all of the paths from start to end vertexes
   - For each of the path P_i, generate the peptide sequence pattern PSP_i
   - Score and rank PSPs by share peaks count score.
3. **Database search by PSP**
   - (details in later part)

Fig. 3. The description of the database search algorithm

Approximate Database Search Using PSP

The matched candidate peptides are searched in the database by PSP. By searching the database, we can find out those protein sequences that have a certain number of matched tagged (with 1 or 2 amino acids errors). But whether there is good match of one peptide sequence in the protein with the whole PSP is not clear. Therefore, it is also a very interesting pattern matching problem.

The approximate matching and pattern matching problem in the context of peptide sequencing is a special matching problem, since it involves both approximate tags matching and approximate masses matching. We have proposed a novel algorithm to solve this novel problem.

The research on string matching has been investigated by many researchers, and the theory and algorithms are quite developed now. It is known that inexact string matching with errors can be done in linear time, and exact string matching with wild-card can be done in linear time [13, 14]. Moreover, the semi-numerical inexact string matching algorithms [13, 14] can be very efficient if the patterns are relatively short. In the PSP algorithm, we have used the semi-numerical inexact string matching algorithms, and the database search process has been done in linear time.

The formal problem definition and the procedure of our algorithm are listed in Fig. 4.

Problem: Approximate database search using PSP

- Input:
  1) *peptide sequence pattern* (PSP)
     $PSP_i = m_1t_1m_2t_2...m_nt_nm_{n+1}$ (m_i and t_i refer to mass and tag, respectively)
  2) database sequence, *Seq*
- Output:
  1) Subsequence Seq_i (or subsequences) in *Seq* that fulfill the requirements
- Requirements:
  1) Approximate match with tags t_i in *Seq* in order, with strict tolerance (every tag with ≤2 amino acids error); if at most $m<n$ tags are present for every database sequences, then these m tags should be approximately matched
  2) Approximate match with masses m_i in *Seq* in order, with loose tolerance (every mass with ≤50 Da mass error)
  3) Efficient process

Procedure: Approximate database search using PSP

1. Select the top PSPs (currently top 3), search database for candidate peptides that approximately match with the tags and masses of these PSPs within certain tolerance.
2. Score and tank the candidate peptides by the share peaks count between their theoretical spectrum and experimental spectrum.
3. Output these peptide sequences.

Fig. 4. Formal description of the approximate pattern matching problem; and the procedure for the PSP algorithm

An illustration of approximate match of PSP to the peptide sequences in the database is in Fig. 5.

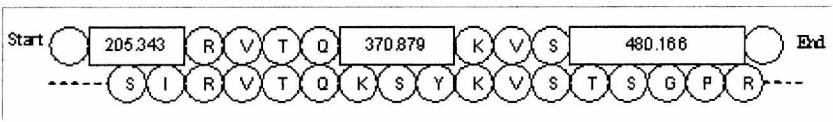

Fig. 5. An example of the match of the peptide sequence pattern (first row) and the peptide sequence in the database (second row)

As illustrated in Fig. 5, the PSP is "[205.343]RVTQ[370.879]KVS[480.166]", with numbers in brackets the mass differences between tags; and the matched peptide sequence is "SIRVTQKSYKVSTSGPR". In this example, the two tags "RVTQ" and "KVS" have matched the identical fragments in the peptide sequence (in other cases, 1 or 2 amino acids mismatches are tolerable). The three mass differences also match with the fragments having similar masses.

As to the running time, for one PSP having length of m and one peptide sequence in database having length n, the algorithm can operate in $O(m+n)$ time. This is much better than the naïve sequence matching method, which requires $O(m*n)$ time. Since there are thousands of peptide sequences in database, the efficiency improvement is very significant. If we load the peptide database into memory once, and search several PSPs against it, the average processing time for one PSP is even shorter.