Connecting X! Tandem to a database management system

Atin Janki
Working Group Databases
and Software Engineering
University of Magdeburg
atin.janki@ovgu.de

Roman Zoun
Working Group Databases
and Software Engineering
University of Magdeburg
roman.zoun@ovgu.de

Kay Schallert
Chair of Bioprocess
Engineering
University of Magdeburg
kay.schallert@ovgu.de

Rohith Ravindran
Working Group Databases
and Software Engineering
University of Magdeburg
rohith.ravindran@ovgu.de

David Broneske
Working Group Databases
and Software Engineering
University of Magdeburg
david.broneske@ovgu.de

Wolfram Fenske
Working Group Databases
and Software Engineering
University of Magdeburg
wolfram.fenske@ovgu.de

Robert Heyer
Chair of Bioprocess
Engineering
University of Magdeburg
robert.heyer@ovgu.de

Dirk Benndorf
Chair of Bioprocess
Engineering
University of Magdeburg
dirk.benndorf@ovgu.de

Gunter Saake
Working Group Databases
and Software Engineering
University of Magdeburg
gunter.saake@ovgu.de

ABSTRACT

Protein identification by mass spectrometry is a valuable method in the field of proteomics and metaproteomics. For protein identification, different protein search engines are used such as X! Tandem, MASCOT, OMSSA, SEQUEST etc. These search engines receive input data in form of files. With the rapid rise of proteomics and metaproteomics, new measurement devices are introduced resulting in increase of research capabilities, consequently producing enormous chunks of data regularly. Admittedly, file-based search engines for protein identification are at their limits and IT methods should be introduced for protein identification to manage huge amount of data efficiently in future. In this paper, we focus on feasibility of Database Management Systems as an alternative to conventional file-based approaches. We implement a connector interface and integrate it into the latest X! Tandem version (2017.02.01) , in order to couple it with a DBMS keeping its business logic intact and study its performance. We compared our work with the core X! Tandem and MetaProteomeAnalyzer tool (which performs protein search and uses a relational database for data storage). We observed there was no information loss in our approach and we were able to successfully implement the DBMS connector interface to X! Tandem.

Categories and Subject Descriptors

H.2 [**Information Systems**]: Protein Identification

General Terms

Design, Performance

Keywords

Data Access, Bioinformatics, Metaproteomics, Proteomics, DBMS

1. INTRODUCTION

Proteomics is the comprehensive study of expressed proteins from one organism for a certain time point; in contrast metaproteomics is the investigation of samples containing proteins from different organisms [1, 2, 3]. Proteomics and metaproteomics use mass spectrometry (MS) as an analytical technique to characterize proteins and detect their accurate masses, which relies upon a protein identification algorithm for cataloging of proteins present in a sample [4]. The protein identification process is based on the study of peptides generated by proteolytic digestion [5, 6]. Algorithms such as X! Tandem [7], MASCOT [8], SEQUEST [9], OMSSA [10] identifies peptides from MS spectra by searching them against a database of known peptides [11, 12, 13]. Balgley et al. [14] found OMSSA and X! Tandem to perform better than SEQUEST and MASCOT with respect to the number of peptide identifications per protein and Quandt et al. [15] in their analysis declared X! Tandem to be more robust than OMSSA and MASCOT when there were changes in the precursor mass error and fragment mass error. Also being an open source software with periodical updates, X! Tandem appears to be a popular choice among biologists.

X! Tandem reads the input data (MS spectra and a protein sequence database) as files and writes the output into a file as well, so any analytical study would require parsing them. The algorithm deals with huge protein libraries (containing over million peptide sequences) and spectra data, which makes it laborious to manipulate and visualize the data as well as the results [16]. Moreover, redundant data tracking and version control is difficult with files. These is-

30^{th} GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 22.05.2018 - 25.05.2018, Wuppertal, Germany.

sues have already been resolved by DBMS. Therefore our project aims to replace the conventional file-based approach with a DBMS. We have implemented a general adapter inside X! Tandem, which can be connected to any DBMS, by keeping its business logic intact and only changing the I/O logic. In this paper we have realized an RDBMS (MySQL) adapter. An RDBMS facilitated us to well represent the underlying relation of input and output data [17, 18].

This paper compares X! Tandem successfully integrated with an RDBMS, the core X! Tandem algorithm and MetaProteomeAnalyzer [19].

Further we discuss basic concepts in the section *Fundamentals*, proposed solution in the section *Our Approach*, followed by *Implementation*, *Evaluation*, and *Conclusion*.

2. RELATED WORK

Zeeberg et al. in their work on GoMiner [20] and Ahmad et al. in their work on nucleolar proteome database [21] have used RDBMS as an efficient storage engine. Yu et al. have realized an RDBMS as a tool for safe warehousing and analysis of quantitative proteomic data [22]. Bjornson et al. have worked towards parallelization of X! Tandem [23] whereas He et al. implemented a parallel X! Tandem with Many Integrated Core (MIC) [24]. Field et al. [25] while working on proteome mass spectral analysis have used RDBMS for storing processed data and customized reporting. MetaProteomeAnalyzer developed by Muth et al. [19], comes closest to our work as they perform protein search using X! Tandem and use RDBMS for storing search results.

3. FUNDAMENTALS

In this section, we explain the basics of a protein search engine with the focus on X! Tandem and briefly about the MPA tool.

3.1 Protein Identification Algorithm

A protein identification algorithm attempts to assign mass spectra to proteins/peptides. Inputs to the algorithm are:

- Protein sequence database (usually found by genetics)
- Experimental spectra (tandem mass spectrometry data usually in MGF[1])
- Configuration parameters

In Figure 1, we show how the experimental spectra relate to the protein sequences in the database.

3.2 Experimental Spectra

Experimental spectra are the result of tandem MS/MS (multiple steps of mass spectrometry, with some form of molecular fragmentation occurring between the stages). These spectra are commonly stored in a MASCOT Generic Format (MGF) file [26] that encodes a collection of spectra. X! Tandem is built to use DTA, PKL or MGF files. We use MGF for our evaluation.

3.3 Protein Sequence Database

Protein sequence database (stored in a file) is a library of known protein sequences that are represented in a standard format [27]. In our work, we used protein sequences stored in a FASTA format file. For every protein sequence in the FASTA file, the first line is the definition line containing an access identifier along with some optional description. The lines following the definition line represent sequence data. The protein search algorithm uses these peptide sequences to create theoretical spectra and matches them with the experimental spectra.

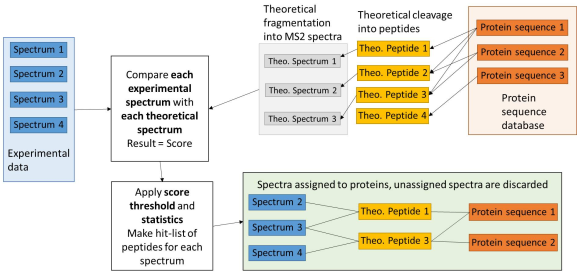
Theoretical fragmentation into MS2 spectra
Theoretical cleavage into peptides
Spectrum 1
Spectrum 2
Spectrum 3
Spectrum 4
Experimental data
Compare each experimental spectrum with each theoretical spectrum Result = Score
Theo. Spectrum 1
Theo. Spectrum 2
Theo. Spectrum 3
Theo. Peptide 1
Theo. Peptide 2
Theo. Peptide 3
Theo. Peptide 4
Protein sequence 1
Protein sequence 2
Protein sequence 3
Protein sequence database
Apply score threshold and statistics Make hit-list of peptides for each spectrum
Spectra assigned to proteins, unassigned spectra are discarded
Spectrum 2
Spectrum 3
Spectrum 4
Theo. Peptide 1
Theo. Peptide 3
Protein sequence 1
Protein sequence 2

Figure 1: Protein search algorithms - General principle

3.4 X! Tandem Output

The output file is in the BIOML (Biopolymer Markup Language) [28] format, which features complex annotations of proteins in a hierarchical manner and can be processed using standard XML parsers.

3.5 MetaProteomeAnalyzer Tool

The MetaProteomeAnalyzer (MPA) tool [19] employs X! Tandem internally with an advanced user interface view. It extracts the MGF and FASTA information from a MySQL DB and converts them into .mgf and .fasta files. Once the protein search is initiated, using these files X! Tandem identifies the proteins and generates the output. The MPA tool then parses the output file and stores it in DB. Hence, it uses both file and DB information for completing the process.

4. OUR APPROACH

With growing size of data it is difficult for biologists to manage hundreds of thousands of files where each file is in gigabytes. Furthermore DBMS have been considered an appropriate and beneficial data storage strategy as they form a classic framework for representing and analyzing huge metaproteomics data [3]. We have seen in subsection 3.5 that the MPA tool stores data in DB but does not read from it directly, during protein identification. Their process of converting data between DB and file representation is inefficient as it introduces an overhead of parsing. Rather than using files if we manage to directly read input from and write output to a DB, it would remove the parsing step, thus reducing load on the entire process of protein identification. Our goal was to design and develop a new architecture for X! Tandem connecting it to a DBMS without altering the protein identification algorithm inside. To store the MGF, FASTA and output files we designed a database schema preserving their hierarchical structure (see Figure 2, 3 and 4). We developed a special adapter interface which could communicate with any database without influencing the functionality of X! Tandem. We used the configuration file input.xml to define the database credentials, MGF and FASTA data source identifier, and parameters.xml to define the calculation cri-

[1] MASCOT Generic Format

teria to match the protein sequences. Other configuration information was kept as a file.

5. IMPLEMENTATION

Our work is implemented in C++ as we have modified X! Tandem classes to read and write data, from and to, MySQL instead of files. We have developed a MySQL adapter interface, which can be modified to connect X! Tandem to any other DB without changing its business logic. Further we study the database design for MGF, FASTA and output files.

6. DATABASE DESIGN

In this section, we discuss the structure of tables for spectra, FASTA and output data in detail.

6.1 Tables for input spectra

MS spectra information is stored into tables: *ms_dataset* and *fragment_ion_list*. While *ms_dataset* stores peptide mass, charge, precursor intensity, retention time (RT) and spectrum title, the peak-list of mass and intensity pairs for each spectrum is stored in *fragment_ion_list* table. Records in *fragment_ion_list* table are mapped to a specific spectrum in *ms_dataset* using a foreign key constraint 'Map_ID' (see Figure 2). Although a join operation on these two tables for reading spectra information would introduce a performance penalty, we do get the flexibility of studying selective spectra as and when required instead of reading the entire file.

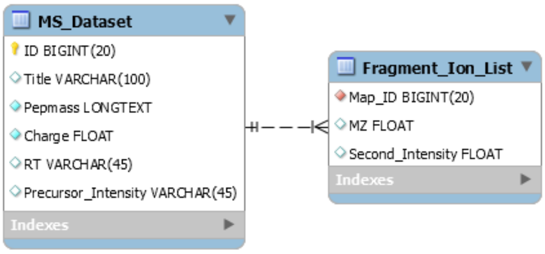

Figure 2: Tables for MS spectra input

6.2 Tables for FASTA input

Understanding its structure (see subsection 3.3), we split each protein sequence into access identifier, description and sequence data and store them in *protein_reference_data*(see Figure 3). The *protein_reference_data_info* table stores the information about the FASTA library loaded into DB.

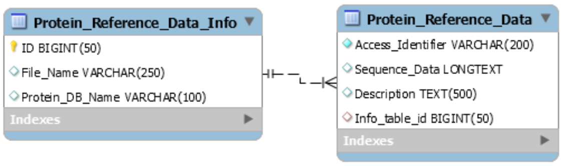

Figure 3: Tables for FASTA input

6.3 Output tables

The X! Tandem output data objects are stored in the tables *out_group* (original mass spectrum), *out_protein* (protein containing matching peptides), *out_domain* (peptide sequences that match to a spectrum), *out_gaml_trace_histograms* (histograms about statistics of an identification), *out_gaml_attributes* (histogram attributes), *out_gaml_xy_data* (histogram values) and *out_parameters_info* (input parameters and performance statistics). The output tables conform to the output standards[2] of core X! Tandem. The complete structure of output tables can be observed in Figure 4.

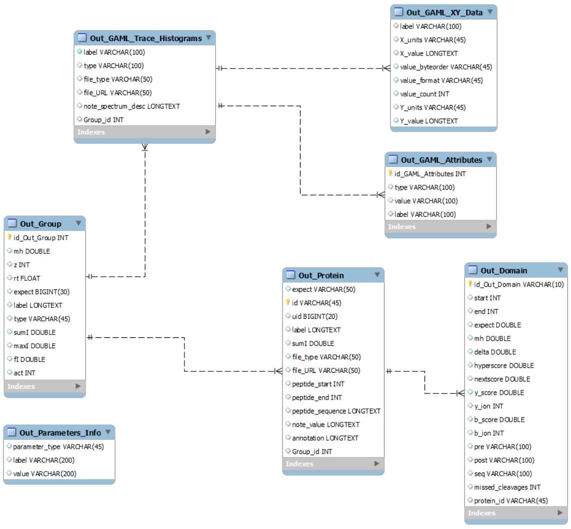

Figure 4: Output tables

7. FACTORY ADAPTER INTERFACE

Factory adapter interface is developed to establish a database connection with X! Tandem. Its implementation only modifies the I/O logic of X! Tandem. The database entities are not coupled with C++ objects of X! Tandem, which means X! Tandem functions without any knowledge of the DB schema. This provides a generic interface where any database can be connected to X! Tandem with changes in input and output schema (pertaining to the DB used) without even worrying about the access and manipulation of data. In our case, we developed a factory adapter interface for MySQL.

8. EVALUATION

We evaluated our work to study the feasibility of integrating X! Tandem with a DBMS with an aim to perform as good as the core X! Tandem. The evaluation was performed on the following hardware:

RAM	: 8GB
Processor	: i5 6th Generation Intel core 2.3 GHz
Operating System	: Windows 10

We conducted experiments with varying sizes of spectra and FASTA data. FASTA datasets used for evaluation-100K_FASTA.fasta and 552K_FASTA.fasta, which contained 100,000 and 552,884 protein sequences respectively were

[2] http://www.thegpm.org/docs/X_series_output_form.pdf

taken from ‘UniProt Knowledgebase’. Spectra datasets used were 100_file.mgf, 2k_file.mgf and 20K_file.mgf which were 100, 2000 and 20000 in spectra counts respectively.

The evaluation was done by assessing the outcomes of all experiments on three performance measures namely *computation time, CPU usage, and RAM usage* for original file-based X! Tandem, the MPA Tool and our approach- X! Tandem using DBMS (MySQL).

For each performance measure, comparing the aforementioned systems, the results were presented in two graphs, one for 100K FASTA and another for 552K FASTA against all the three datasets of spectra. Consequently we verified them and concluded that there was no information loss from our approach.

8.1 Computation time

For small-sized input data (100 spectra with 100K, 552K FASTA and 2000 spectra with 552K FASTA) our work (8.48, 24.67 and 32.34 seconds) outperforms the core X! Tandem (9.06, 46.56 and 73.25 seconds). For 2000 spectra with 100K FASTA our approach (32.34 seconds) was slightly slower than the core X! Tandem (23.67 seconds). However instead for input spectra of size 20K with 100K and 552K FASTA, our approach (606.06 and 1168.33 seconds) was considerably slower than core X! Tandem (185.34 and 449.94) as it takes almost 3 times more time to execute. To deal with this issue, batch processing of data should be included in our approach. In comparison to the MPA tool, our approach performs significantly better in all cases (see Figure 5 and 6.

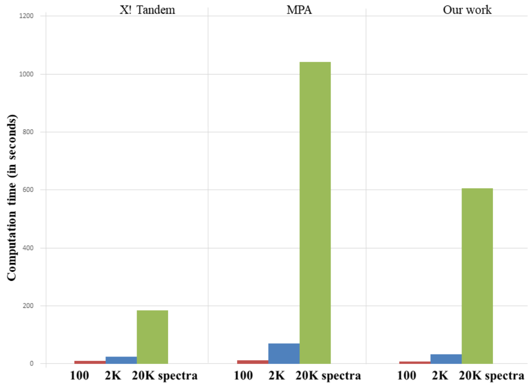

Figure 5: Computation Time Comparison - 100k FASTA with Spectra up to 20K

8.2 CPU Usage

We studied CPU usage of the three systems when no other process was running on the machine. We noticed that CPU usage is remarkably less for our approach (varying from 8.88 to 17.95%) irrespective of the size of data whereas in case of core X! Tandem and the MPA tool, CPU usage varies from 71.69% to 100% and 85-100% respectively (see Figure 7 and 8). Higher CPU usages could lead to performance issues in the system.

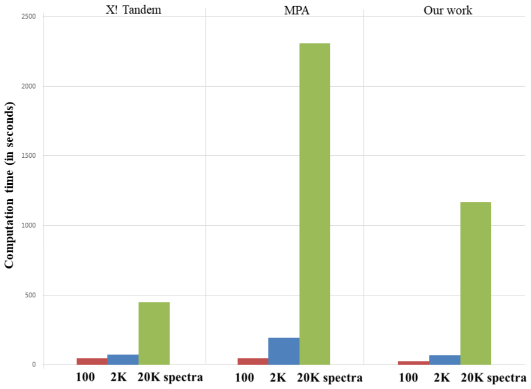

Figure 6: Computation Time Comparison - 552k FASTA with Spectra up to 20K

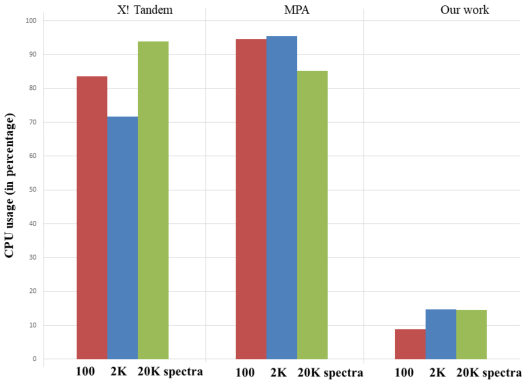

Figure 7: CPU Usage Comparison - 100k FASTA with Spectra up to 20K

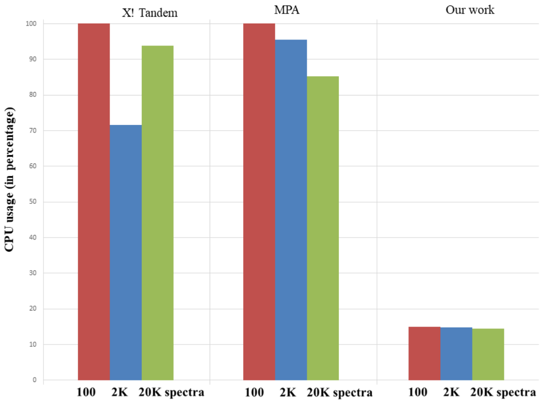

Figure 8: CPU Usage Comparison - 552k FASTA with Spectra up to 20K

8.3 RAM Usage

We can observe from Figure 9 and 10 that RAM usage is comparatively same in all the systems for small-sized input data (100 spectra with 100K & 552K FASTA) with core X! Tandem, MPA and our work having 66.69 & 190.61, 56.96 & 177.54, 54.48 & 248.04 bytes consumption respectively. However, our approach consumes significantly more amount of RAM (2429.94 & 2974.06 bytes) for large input data (20K spectra with 100K/552K FASTA) against that of core X! Tandem (237.94 & 392.83 bytes) and the MPA tool (47.93 & 177.34 bytes). RAM consumption increases linearly with data size, in our case. The MPA tool recorded lowest RAM consumption in all the cases.

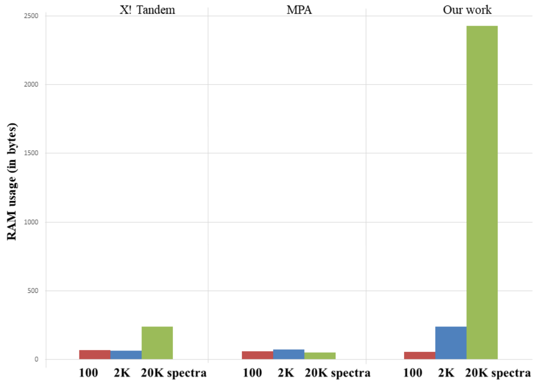

Figure 9: RAM Usage Comparison - 100k FASTA with Spectra up to 20K

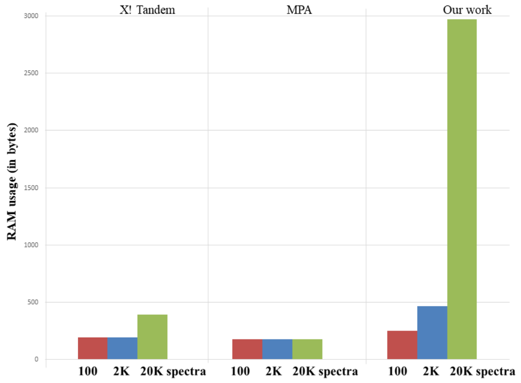

Figure 10: RAM Usage Comparison - 552k FASTA with Spectra up to 20K

The evaluation results show that core X! Tandem is the fastest as it is highly optimized. Our approach was noted to be faster than core X! Tandem while dealing with small-sized data whereas for larger data it was almost 3 times slower, further drawing our attention to a necessary implementation of batch processing. Our approach was quicker than the MPA tool in all the cases. However our approach exhibited efficient CPU usages across all the experiments, outshining the other two systems by a wide margin. In terms of RAM usage, our approach needs improvement as it consumed a lot more memory than the other two systems when data size increased.

9. CONCLUSION

We have not only engineered a connector interface between X! Tandem and a DBMS but also systematically investigated the feasibility of moving from file-based protein search algorithm to DBMS based algorithm without any information loss. We observed that DBMS offers accessibility to data in a structured manner that was much needed for biologists. A biologist may create SQL queries on results to create customized reports without going through the hassle of parsing the files. Also in file-based approach, FASTA data was separated with respect to taxon, in different files. However with a connection to DBMS, all the FASTA data could be stored in one database and could be selectively used for experiments.

During evaluation we observed core X! Tandem to be the fastest of the three systems as it is highly optimized. Our work was faster than core X! Tandem for small datasets but needed batch processing for handling large datasets efficiently. We were significantly faster than MPA in all the cases. There was no overhead noticed on database access in our approach for small-sized input spectra, but a drastic overhead was noticed for large input spectra. This implies our approach needs multi-threading for cost-effective RAM usage. Our approach exhibited efficient CPU usages across all the experiments, outshining the other two systems by a wide margin.

We have successfully developed an adapter to connect X! Tandem to any database (Section 7), opening up many possibilities for future improvements. For instance, an implementation of NoSQL database using our approach would provide an easy scale-out architecture with efficient performance whereas file-based X! Tandem could not scale. Also our work provides a basis for realizing protein identification algorithms in cloud environments while utilizing features of BigData.

10. FUTURE WORK

Our connector interface for MySQL could be exchanged (Section 7) for cloud-based endpoints such as Cassandra. Such cloud-based endpoints provide elastic scalability, high availability and fault tolerance with high performance. That way protein identification could be developed as a service, which would bring an effective way of collaboration amongst biologists because of its central storage. Multi-threading approach should be adopted to tackle high RAM usage in our work.

11. ACKNOWLEDGEMENT

The authors sincerely thank Xiao Chen, Sebastian Krieter, Andreas Meister and Marcus Pinnecke for their support and advice. This work is partly funded by the de.NBI Network (031L0103), the DFG (grant no.: SA 465/50-1), the European Regional Development Fund (grant no. 11.000sz00.00.0 17 114347 0, the German Federal Ministry of Food and Agriculture (grants nos. 22404015) and dedicated to the memory of Mikhail Zoun.

12. REFERENCES

[1] R. Pieper, S.-T. Huang, and M.-J. Suh, "Proteomics and metaproteomics," in *Encyclopedia of Metagenomics*. Springer New York, 2013, pp. 1–11.

[2] R. Heyer, F. Kohrs, U. Reichl, and D. Benndorf, "Metaproteomics of complex microbial communities in biogas plants," *Microbial Technology*, vol. 8, 04 2015.

[3] R. Heyer, K. Schallert, R. Zoun, B. Becher, G. Saake, and D. Benndorf, “Challenges and perspectives of metaproteomic data analysis,” *Journal of Biotechnology*, vol. 261, no. Supplement C, pp. 24 – 36, 2017, bioinformatics Solutions for Big Data Analysis in Life Sciences presented by the German Network for Bioinformatics Infrastructure.

[4] R. Aebersold and M. Mann, “Mass spectrometry-based proteomics,” *Nature*, vol. 422, no. 6928, p. 198, 2003.

[5] M. W. Duncan, R. Aebersold, and R. M. Caprioli, “The pros and cons of peptide-centric proteomics,” *Nature Biotechnology*, 2010.

[6] J. Eriksson and D. Fenyö, “Modeling mass spectrometry-based protein analysis,” *Bioinformatics for Comparative Proteomics*, pp. 109–117, 2011.

[7] R. Craig and R. C. Beavis, “Tandem: matching proteins with tandem mass spectra,” *Bioinformatics*, vol. 20, no. 9, pp. 1466–1467, 2004.

[8] J. S. Cottrell and U. London, “Probability-based protein identification by searching sequence databases using mass spectrometry data,” *electrophoresis*, vol. 20, no. 18, pp. 3551–3567, 1999.

[9] J. K. Eng, A. L. McCormack, and J. R. Yates, “An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database,” *Journal of the American Society for Mass Spectrometry*, vol. 5, no. 11, pp. 976–989, 1994.

[10] L. Y. Geer, S. P. Markey, J. A. Kowalak, L. Wagner, M. Xu, D. M. Maynard, X. Yang, W. Shi, and S. H. Bryant, “Open mass spectrometry search algorithm,” *Journal of proteome research*, vol. 3, no. 5, pp. 958–964, 2004.

[11] L. J. Everett, C. Bierl, and S. R. Master, “Unbiased statistical analysis for multi-stage proteomic search strategies,” *Journal of proteome research*, vol. 9, no. 2, pp. 700–707, 2010.

[12] M. V. Ivanov, L. I. Levitsky, and M. V. Gorshkov, “Adaptation of decoy fusion strategy for existing multi-stage search workflows,” *Journal of The American Society for Mass Spectrometry*, vol. 27, no. 9, pp. 1579–1582, 2016.

[13] R. D. Bjornson, N. J. Carriero, C. Colangelo, M. Shifman, K.-H. Cheung, P. L. Miller, and K. Williams, “X!! tandem, an improved method for running x! tandem in parallel on collections of commodity computers,” *The Journal of Proteome Research*, vol. 7, no. 1, pp. 293–299, 2007.

[14] B. M. Balgley, T. Laudeman, L. Yang, T. Song, and C. S. Lee, “Comparative evaluation of tandem ms search algorithms using a target-decoy search strategy,” *Molecular & Cellular Proteomics*, vol. 6, no. 9, pp. 1599–1608, 2007.

[15] A. Quandt, L. Espona, A. Balasko, H. Weisser, M.-Y. Brusniak, P. Kunszt, R. Aebersold, and L. MalmstrÃűm, “Using synthetic peptides to benchmark peptide identification software and search parameters for ms/ms data analysis,” *EuPA Open Proteomics*, vol. 5, pp. 21 – 31, 2014.

[16] R. Zoun, K. Schallert, D. Broneske, R. Heyer, D. Benndorf, and G. Saake, “Interactive chord visualization for metaproteomics,” in *2017 28th International Workshop on Database and Expert Systems Applications (DEXA)*, Aug 2017, pp. 79–83.

[17] C. Türker and G. Saake, “Objektrelationale datenbanken: Ein lehrbuch. 1,” *Auflage. Heidelberg: dpunkt. verlag GmbH*, 2006.

[18] G. Saake, K. Sattler, and A. Heuer, “Datenbanken-konzepte und sprachen, mitp professional, 2013.”

[19] T. Muth, A. Behne, R. Heyer, F. Kohrs, D. Benndorf, M. Hoffmann, M. LehtevÃd’, U. Reichl, L. Martens, and E. Rapp, “The MetaProteomeAnalyzer: A powerful open-source software suite for metaproteomics data analysis and interpretation,” *Journal of Proteome Research*, vol. 14, no. 3, pp. 1557–1565, feb 2015.

[20] B. R. Zeeberg, W. Feng, G. Wang, M. D. Wang, A. T. Fojo, M. Sunshine, S. Narasimhan, D. W. Kane, W. C. Reinhold, S. Lababidi *et al.*, “Gominer: a resource for biological interpretation of genomic and proteomic data,” *Genome biology*, vol. 4, no. 4, p. R28, 2003.

[21] Y. Ahmad, F.-M. Boisvert, P. Gregor, A. Cobley, and A. I. Lamond, “Nopdb: Nucleolar proteome database,” *Nucleic Acids Research*, vol. 37, no. 1, pp. D181–D184, 2009.

[22] K. Yu and A. R. Salomon, “Peptidedepot: flexible relational database for visual analysis of quantitative proteomic data and integration of existing protein information,” *Proteomics*, vol. 9, no. 23, pp. 5350–5358, 2009.

[23] R. D. Bjornson, N. J. Carriero, C. Colangelo, M. Shifman, K.-H. Cheung, P. L. Miller, and K. Williams, “X!! tandem, an improved method for running x! tandem in parallel on collections of commodity computers,” *The Journal of Proteome Research*, vol. 7, no. 1, pp. 293–299, 2007.

[24] P. He and K. Li, “Mic-tandem: parallel x! tandem using mic on tandem mass spectrometry based proteomics data,” in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2015.

[25] H. I. Field, D. Fenyo, and R. C. Beavis, “Radars, a bioinformatics solution that automates proteome mass spectral analysis, optimises protein identification, and archives data in a relational database,” *Proteomics*, vol. 2, no. 1, p. 36, 2002.

[26] “Mascot generic format documentation.” [Online]. Available: http://www.matrixscience.com/help/data_file_help.html

[27] N. C. for Biotechnology Information. (2002, Nov.) Fasta format. [Online]. Available: https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=BlastHelp

[28] D. FenyâĹŽâĹĆ, “The biopolymer markup language.” *Bioinformatics (Oxford, England)*, vol. 15, no. 4, pp. 339–340, 1999.