2009 11th IEEE International Conference on High Performance Computing and Communications

QuantWiz: A Parallel Software Package for LC-MS-based Label-free Protein Quantification

Jing Wang[1], Yunquan Zhang[1, 2], Xianyi Zhang[1], Xiangzheng Sun[1, 2, 3], Zelin Hu[1, 2, 3], Sujun Li[4], Rong Zeng[4]

[1]*Laboratory of Parallel Computing, Institute of Software, Chinese Academy of Sciences*

[2]*State Key Laboratory of Computer Science, Chinese Academy of Sciences*

[3]*Graduate University of Chinese Academy of Sciences*

[4]*Key Laboratory of Systems Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences*

{wangj, zyq, zxy, sxz, hzl}@mail.rdcps.ac.cn, {sjli, zr}@sibs.ac.cn

Abstract

Nowadays Proteomics becomes more and more popular in life science. Protein quantification, especially based on mass spectrometry (short for MS) method, is perceived as an essential part of research on proteomics. There have been some algorithms and software for protein quantification based on MS. But they have difficulties on portability, applicability and longtime running. To solve these problems, we developed a new domestic parallel software package called QuantWiz for high performance Liquid Chromatography (short for LC)-MS-based label-free protein quantification. In this paper, we described the framework design and prototype development of this high performance software package firstly. Also, user interface developed for the visualization of QuantWiz is introduced. Finally, we showed implementation of the parallelization version and performance of some experiments on this software package.

Keywords: *protein quantification, mass spectrometry, label-free, abundance ratio, parallelization*

1. Introduction

As the post-genomic time arrived, proteomics becomes one of the most important research fields in life science. Biological functions of proteins not just depend on their specific 3D (three-dimension) structures, but also rely on the variation of the quantity. Therefore protein quantification is one of the critical procedures in the researching progress. Recently, quantification based on mass spectrometry (short for MS) turns into a core subject for many researchers [1]. After high throughput proteomics experiment based on MS, processing and analyzing of these large amounts of data is obviously of great importance.

Based on the information inherent in chromatographic data, MS spectra, and MS-MS (MS2, or shotgun tandem mass spectrometry) -based peptide assignments, label-free quantitative strategies are attractive alternatives for quantitative LC-MS-MS-based proteomics because of their simplicity, affordability, and flexibility. Since label-free quantification has emerged as a promising technology for proteome analysis, computational methods are required for the accurate extraction of peptide signals from LC-MS data and the tracking of these features across the measurements of different samples.

There have been some algorithms and software such as TPP [2], OpenMS [3], SuperHirn [4], MapQuant [5] and ASAPRatio [6] and so on for protein quantification based on mass spectrometry. However, these algorithms and software all have the problem of portability. Some are just for isotope-labeled protein quantification, such as ASAPRatio. Besides, some do not support data format from specific mass spectrometer, for example, high-precision data from Shanghai Institutes of Biological Sciences (short for SIBS below). Moreover, running time of some software as ASAPRatio is too long and all those algorithms and software mentioned above have no parallel version for high throughput data processing.

Before starting our research work on QuantWiz, we have intensively studied all these software packages mentioned above. Taking into account the problem emerged in other algorithms or software, we designed the framework and developed QuantWiz for high

978-0-7695-3738-2/09 $25.00 © 2009 IEEE
DOI 10.1109/HPCC.2009.62

performance LC-MS-based label-free protein quantification. It has features as follows:

1) QuantWiz is developed by JAVA to solve the problem of portability.

2) This software package supports three kinds of data formats. First, it supports the standard mzXML mass spectrometry data format [7]. The pepXML format [8], which is XML format for storage, exchange, and processing of peptide sequences derived from ms/ms scans, is also supported for the identified peptides. Last but most importantly, the specified format defined by tab-delimiter of SIBS is supported by QuantWiz.

3) We applied some optimization techniques to QuantWiz to reduce the execution time in other algorithms or software. We use data blocks and B-tree index structure to speed up data querying efficiency.

4) Parallelization was performed in some experiments subsequently.

This paper is organized as follows. The software design of QuantWiz will be introduced at first, which includes software architecture and parallelization method. Afterwards we evaluate on both accuracy and scalability of the performance. Ultimately, we give out the conclusion.

2. QuantWiz Software Design

2.1. QuantWiz Software Architecture

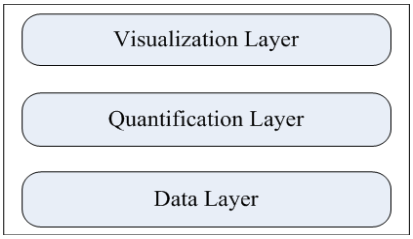

Figure 1 Software Architecture of QuantWiz

QuantWiz is designed as three layers and its architecture is shown in figure 1. The three layers are the data layer, with the quantification layer in the middle, and visualization layer on the top. Each layer has its own function:

1) Data layer: creating file index, maintaining data buffer and managing MS data. The data layer is a buffered storage manager for indexed LC-MS data, in which the LC-MS data is divided into blocks and a simple and robust B-tree index structure is constructed. This index will speed up data querying efficiency dramatically.

2) Quantification layer: defining abstract classes, interfaces and each module in the process of protein quantification. We will discuss this in detail in Section 2.2.

3) Visualization layer: menu for users' visualization. This part will be described in Section 2.3.

2.2. Quantification Process

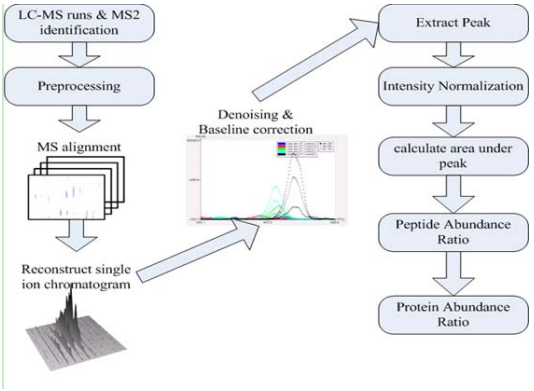

Figure 2 Quantification Process of QuantWiz

The quantification process, which is shown in figure 2, is the core part of this software package. When we get the LC-MS runs or MS2 data, we perform preprocessing to extract features on the input LC-MS raw data and MS2 identifications which are associated with detected features from a database search. In order to create an alignment topology, MS-alignments are performed for every LC-MS pair. Thus we can reconstruct the single ion chromatogram. After the operation of denoising and baseline correction, we extract peak prepared for the calculation of peptide abundance ratio. But before that, intensity normalization needs to be done to avoid experimental artifacts as differences in the loading volumes between samples, peptide ionization variations or decouple feature intensities from the original peptide concentrations.

The "abundance ratio" mentioned above is a critical concept of protein quantification. It is what we need as the final result. An abundance ratio in this paper means the relative quantitative ratio in different samples for the same peptide or protein. The peptide abundance ratio is calculated from the area under peak. However, this is not the final result. We will discuss in detail how to infer protein abundance ratio from peptide abundance ratio later.

Note that different algorithms have different effects and advantages for different data. Therefore, we provide multiple algorithm modules in some parts of our process for users to choose, and consequently increase scalability and flexibility.

Table 1 Optional Algorithm modules in QuantWiz

Interface	Algorithm Module
MS alignment	Polynomial Fitting
Reconstruct SIC	Extract Single Isotope Peak
Denoising	SG Smoothing
Denoising	BMIE based on Wavelet
Baseline correction	Lowess Smoothing
Extract Peak	Find LC valley
Extract Peak	Extract LC Peak based on Maximum Value
Intensity Normalization	Global Normalization
Calculate AUP	Weighted Average of original LC Peak Area and Denoised Peak Area

From the above workflows shown in figure 2, we can get the abundance ratios of peptides. However, this is not enough, because the final results we expect are protein abundance ratios. We called the process from peptides to proteins as "protein inference". Figure 3 shows us how we get protein abundance ratio from peptide abundance ratio. First, multiple peptide abundance ratios for the same peptide were clustered into several classes by Xmean method [9] from WEKA package based on JAVA. The reason we chose this method is that it can automatically give optimized number of clusters. We chose the class with best correlation coefficient as a unique peptide abundance ratio. Second, with the correlation coefficient as weight, the average of unique peptide abundance ratios was calculated as the final protein abundance ratio.

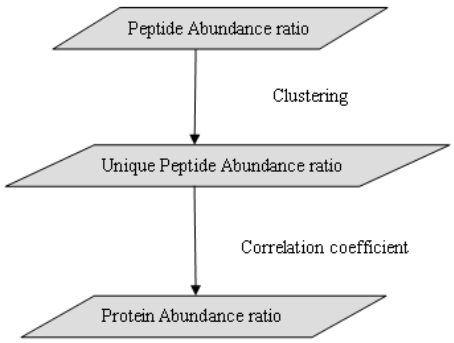

Figure 3 Protein Inference Flow Paths

2.3. Visualization

For user friendly, we developed visualization interface as a part of QuantWiz software package. An example of the use of this interface is displayed in figure 5. It has several features:

1) Support standard MS data format and special format of SIBS.

2) Support 2D (two-dimension) and 3D (three-dimension) display of LC graphs, as shown in figure 5.

3) Function of simple manually quantification.

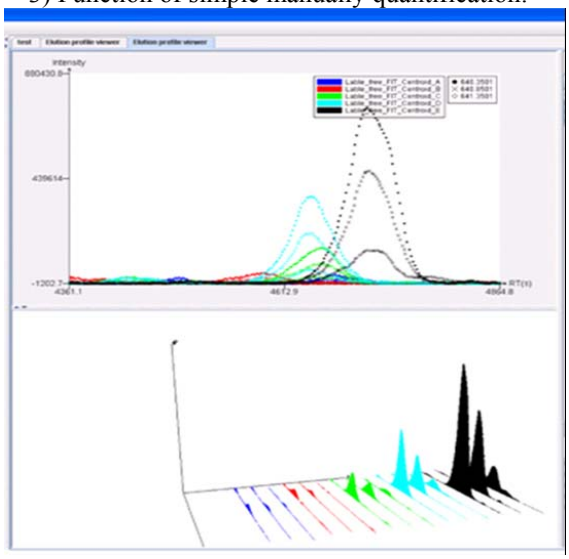

Figure 4 2D and 3D Display of LC Graph

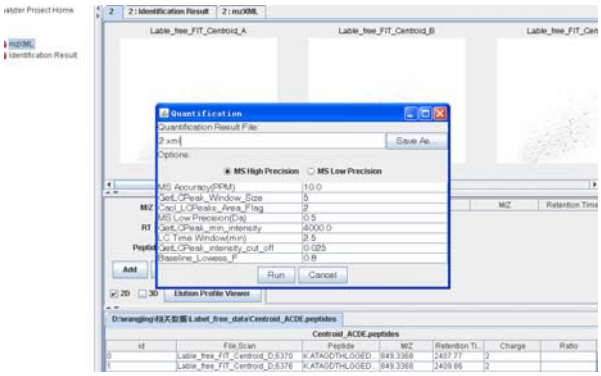

Figure 5 User Interface of QuantWiz

For sake of clear view, we just show the main part of the user interface of QuantWiz in figure 5. On the left is the file tree view. Peptide identification data, that is MS data, is on the top. Specific MS data can be found by the search according to its special attribute values in the middle. At the bottom shows each MS data and final result will be displayed there. We can get the abundance ratios after quantification process showed in the top view in figure 5.

2.4. Parallelization

We chose shared memory paradigm (short for SMP) to parallel in our algorithm. With data parallelization,

we realize the parallelization work in the quantification process shown in figure 2. Specifically, we used main thread and thread pool with slave threads in JAVA as indicated in figure 6. The reason we used thread pool is that slave thread minimizes the overhead due to thread creation. As we know, thread objects use a significant amount of memory. In a large-scale application such as in our work, allocating and deallocating many thread objects create a significant memory management overhead. Yet in thread pool, thread has been ready when requested, thus reduce the overhead of thread creation. In our work, each of the two kinds of threads has its own function respectively as following:

1) Master Thread: reading and dispatching MS data.

2) Slave Threads: processing data and performing the quantification.

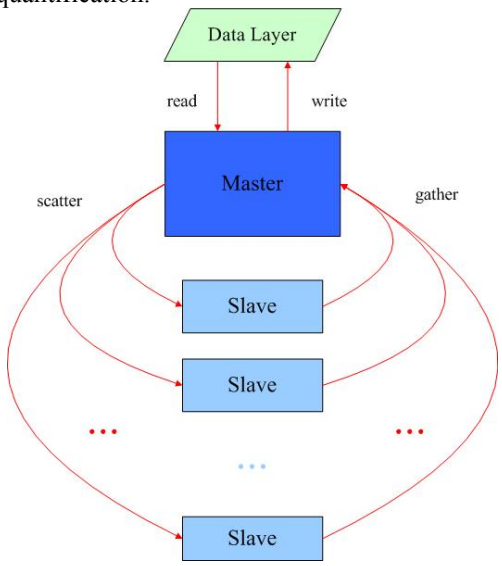

Figure 6 Parallelization Method of QuantWiz

3. Software Evaluation

3.1. Quantification Accuracy

We used the standard testing dataset provided by SIBS, with 5 MS runs and 325 peptide identification results. The expected abundance ratio is 1:1:2:5:10. After the quantification of QuantWiz, we got 79 unique peptide abundance ratios and finally 49 protein abundance ratios.

For the first two samples in figure 7, quantification results from QuantWiz are rather close to the standard ratios, but in the other three samples, the error becomes larger. Anyway, the trends are correct between the results from QuantWiz and expected values, yet the precision needs to be improved especially when abundance ratios are larger.

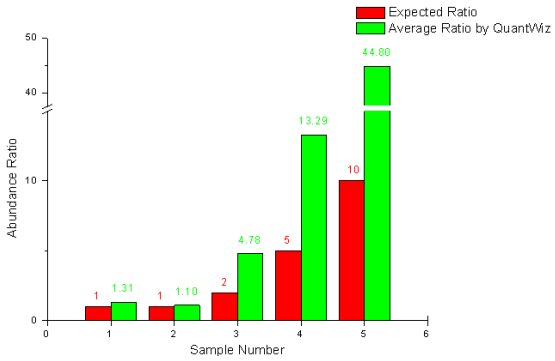

Figure 7 Comparisons of Expected Ratio and Average Ratio by QuantWiz

3.2. Parallel Performance

The platform we performed evaluation is a SMP server of 4-way Quad-Core processors:

1) Processors: AMD Opteron 8354/2.2GHz

2) Memory: 16GB

3) Operating System: Ubuntu 8.10

4) JAVA Runtime Environment 1.6

The MS data we used is from Orbitrap mass spectrometer provided by SIBS, with 12 MS runs and 88836 peptides. Because of the condition of hardware platform, we have only evaluated the parallel performance once. But according to what we discussed before, we can see that the execution time will not vary greatly under multiple experiments. As given in figure 8, execution time using 1 thread is 79637 seconds, that is, more than 22 hours which is nearly a whole day. On the contrary, task can be finished in 5412 seconds which is 1.5 hours if we use 16 threads (Figure 8). So the speedup is 14.71 on the 16-core server when thread number is 16 which are shown in Figure 9.

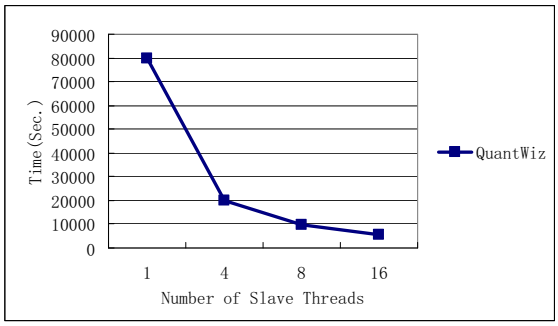

Figure 8 Excution Time Using Different Numbers of Thread

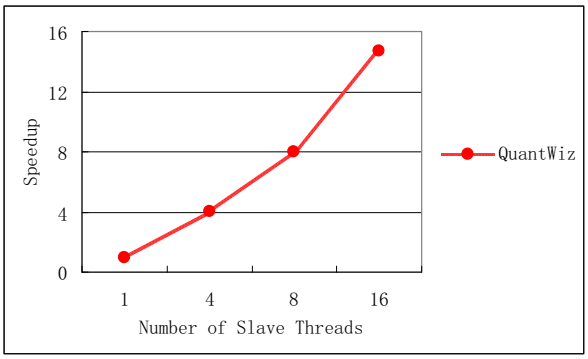

Figure 9 Speedup of QuantWiz Using Different Numbers of Thread

4. Conclusion

In our work, we designed the framework and the developed the software package QuantWiz for LC-MS-based label-free protein quantification. Using our performance optimization techniques, several difficulties as portability, applicability, long running time that existed in some other popular algorithms or software has been avoided. We also implemented the parallelization of QuantWiz and performed some experiments on it.

Meanwhile, there is some works needed to be continued in the future. First of all, quantification precision in our software package is not as well as expected. This will be a main work for us later on. Further discussion and analysis of users' requirement of SIBS is also needed in the future.

5. Acknowledgment

This work is supported in part by CAS Project No.KGCX1-YW-13, the National Natural Science Foundation of China under Grant No.60303020, Key Program of National Nature Foundation of China under Grant No. 60533020 and the National High-Tech Research and Development Plan of China ("863" plan) under Grant Nos.2006AA01A102 and 2006AA01A125.

We are grateful to Associate Professor Quanhu Sheng from Shanghai Institutes of Biological Sciences for his professional help and valuable suggestion. Thanks Yuxin Tang and Shengfei Liu from Laboratory of Parallel Computing, Institute of Software, CAS, for their helpful discussion.

Reference

[1] R. Aebersold, M. Mann. Mass spectrometry-based proteomics. *Nature*, 2003, 422:198-207.

[2]http://tools.proteomecenter.org/wiki/index.php?title=Software:TPP

[3]http://tools.proteomecenter.org/wiki/index.php?title=Software:SuperHirn

[4] http://sourceforge.net/projects/open-ms/

[5] http://arep.med.harvard.edu/MapQuant/

[6] X.J. Li, H. Zhang, et al. Automated Statistical Analysis of Protein Abundance Ratios from Data Generated by Stable-Isotope Dilution and Tandem Mass Spectrometry. *Analytical Chemistry*, 2003, 75(23):6648-6657.

[7] P.G.A. Pedrioli, J.K. Eng, R. Hubley, et al.. A common open representation of mass spectrometry data and its application to proteomics research, *Nat Biotech*, 2004, 22: 1459-1466.

[8] A. Keller, J. Eng, N. Zhang, et al. A uniform proteomics MS/MS analysis platform utilizing open XML file formats, *Mol Syst Biol*, 2005, 1:2005.0017

[9] I.H. Witten, E Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers: Hamilton, 2000.