Tandem Mass Spectrometry Protein Identification on a PC Grid

D. Zosso[a], M. Podvinec[a,1], M. Müller[b], R. Aebersold[b], M.C. Peitsch[c], T. Schwede[a]
[a] *Swiss Institute of Bioinformatics and Biozentrum, University of Basel, Switzerland*
[b] *Institute of Molecular Systems Biology, ETHZ, Switzerland*
[c] *Novartis Institutes of BioMedical Research, Basel, Switzerland*

Abstract. We present a method to grid-enable tandem mass spectrometry protein identification. The implemented parallelization strategy embeds the open-source x!tandem tool in a grid-enabled workflow. This allows rapid analysis of large-scale mass spectrometry experiments on existing heterogeneous hardware. We have explored different data-splitting schemes, considering both splitting spectra datasets and protein databases, and examine the impact of the different schemes on scoring and computation time. While resulting peptide e-values exhibit fluctuation, we show that these variations are small, caused by statistical rather than numerical instability, and are not specific to the grid environment. The correlation coefficient of results obtained on a standalone machine versus the grid environment is found to be better than 0.933 for spectra and 0.984 for protein identification, demonstrating the validity of our approach. Finally, we examine the effect of different splitting schemes of spectra and protein data on CPU time and overall wall clock time, revealing that judicious splitting of both data sets yields best overall performance.

Keywords. Proteomics, Tandem MS, Protein Identification, PC Grid.

Introduction

Proteomics is the systematic, parallel study of ensembles of proteins found in particular cell types or present under particular exogenous or endogenous conditions. These studies promise deep and detailed insights into the control and function of biological systems by comparing samples from different tissues, developmental stages or disease states [1]. Along with other high-volume experimental techniques, proteomics is a cornerstone of systems biology, an emerging discipline characterized by the systematic and quantitative large-scale collection of data, linked to the use of computational biology to model and predict the behavior of complex biological systems [2].

Proteomics and systems biology are increasingly applied in clinically relevant fields of research, such as the biology of cancer, diabetes or other multifactorial diseases [3, 4]. Moreover, they are vital tools in the development of new drugs [5] and the study of biomarkers (i.e. disease-related alterations of the protein composition of accessible body fluids) in the diagnosis of diseases, the adaptation of therapy to inter-individual variation in drug metabolism, response, and toxicity. [6-9].

[1] Corresponding Author: Swiss Institute of Bioinformatics, University of Basel, Klingelbergstr. 50-70, CH-4056 Basel; phone: +41 61 267 15 83, fax: +41 61 267 15 84. michael.podvinec@unibas.ch

To identify proteins present in a particular biological sample, tandem mass spectrometry (MS/MS) combined with bioinformatics analysis is commonly applied. Briefly, the following steps are performed: First, the proteins contained within a biological sample are separated into fractions along one or multiple dimensions, such as size, isoelectric point, or hydrophobicity. Subsequently, fractions are digested by specific proteases, ionized and injected into a mass spectrometer, where peptide parent ions are selected, fragmented, and the mass fingerprint of the fragments is acquired.

Bioinformatics approaches for protein identification are computationally expensive (see [10, 11]). Observed peptide fragment masses are compared against peptide mass fingerprints computed from a database of protein target sequences. To model peptide fingerprints, the protein database must be expanded into a list of expected peptides by chopping sequences at specific proteolytic cleavage sites. This step models protein digestion prior to MS/MS analysis. Further database expansion criteria may include coverage of missed or unanticipated cleavage sites, mutations of single amino acids, and constant or potential amino acid mass modifications. The last can occur during sample preparation or may be post-translational modifications (PTM) of biological relevance. Expansion of peptide mutations and of a list of potential PTMs must be exhaustive, since completeness of annotated mutations and PTMs is not guaranteed in protein sequence databases. As all these expansion criteria are orthogonal, their joint extension rapidly leads to combinatorial explosion of the computational complexity.

A single MS/MS experiment routinely results in tens of thousands of spectra, e.g. in [12], which leads to considerable computational requirements for analysis and forces strict limits on expansion criteria to prevent the search problem from becoming intractable. On the other hand, experiments typically yield a significant proportion of spectra that cannot be assigned to a peptide in spite of good data quality. It therefore makes sense to allow for as many biologically meaningful peptide modifications as possible in an attempt to match previously unidentified spectra. Such searches require the availability of large computational resources, as can be provided by a compute grid. Moreover, protein identification from MS/MS spectra is essentially a data-parallel problem, and is well suited for efficient grid-based execution.

In this paper, we demonstrate how x!tandem [13], a publicly available MS/MS identification tool can be grid-enabled. Our system allows users to submit MS/MS data for analysis via a web front end. Protein identification is performed on a grid of hundreds of desktop PCs. This work extends the *ParallelTandem* parallelization strategy [14] and adapts it to a grid environment. In particular, we have investigated the impact of parallelization on search results in terms of their numerical stability, the stability of the score statistics, detection characteristics (sensitivity and specificity), and runtime.

1. Materials and Methods

1.1. Tandem Mass Spectrometry Protein Identification Tool

X!tandem is an open-source implementation of an algorithm to match a set of peptide tandem mass spectra with a list of protein sequences [13, 15]. The free availability of the executable, its source code and plug-ins make this tool particularly attractive for (academic) grid environments, where licensing costs and models, flexibility, maintainability and portability are important issues. Its pluggable architecture allows the integration of additional scoring schemes.

X!tandem splits the matching process into two sequential steps. First, spectra are assessed against the complete database of proteins with a low level of model complexity, permitting rapid elimination of non-matching sequences and establishing a set of candidate proteins. On these candidates, a second, refined search is carried out, resulting in additional peptide identifications.

At the beginning of both the non-refined and refined search, the protein sequences are expanded into a peptide list. For each peptide, the mass values of its possible fragment ions are calculated, producing an artificial peptide mass fingerprint. This fingerprint is compared to each measured tandem mass spectrum and scored using either the native or a plugged-in scoring scheme.

The native x!tandem scoring scheme, called hyperscore, is based on the dot-product between spectrum (*I*) and prediction (*P*) peaks (eq. 1). N_b and N_y are the number of matched b- and y-ions, respectively, i.e. N- and C-terminal peptide fragments.

$$Hyperscore = N_b!\,N_y!\sum_i I_i P_i \qquad (1)$$

In order to assess the statistical significance of a calculated hyperscore, x!tandem computes an expectation value (e-value) as proposed in [16]. For each peptide, a hyperscore histogram of all scored spectra is established. Only the highest-scoring spectrum is supposed to be a valid match and all other spectra are considered as random matches. The p-value of the valid score, i.e. the probability of observing that score at random, can be estimated by log-linear extrapolation of the right-hand tail of this extreme value distribution. Multiplying this value by the number of scored sequences yields the expected number of equal scores, considering the given set of spectra and peptide list.

Once the peptide evidence is established, x!tandem attempts to infer protein identities. Based on the number of peptide hits *n* of a protein and their respective scores e_i, a protein e-value is calculated according to eq. 2:

$$e_{protein} = \binom{s}{n} \cdot \frac{p^n (1-p)^{s-n}}{sN^{n-1}} \cdot \prod_{i=1}^{n} e_i \qquad (2)$$

where *s* is the number of mass spectra in the dataset, *N* the number of peptide sequences scored to find the unique peptides, and *p* is *N* divided by the total number of peptides in the considered protein expansion. The equation is a Bayesian model for a protein to have obtained the observed number of matches by chance. The first two terms mainly describe the probability of random parent mass matches, a concept also suggested in [17], whereas the product of the underlying peptide e-values takes into account their non-randomness with respect to their fragment mass fingerprints.

1.2. Parallelization Scheme

X!tandem supports multithreading on suitable machine architectures. In a cluster environment, however, parallelization requires inter-node communication tools like PVM or MPI. Such a cluster-based parallelization strategy has been implemented in *ParallelTandem* [14]. The authors introduce the distribution of subsets of the initial mass spectra to different cluster nodes. After a first non-refined step, results are collected, candidate proteins are extracted from all subjobs, and refinement jobs are sent out to the nodes again. A consolidation step calculates the final protein e-values.

A major constraint in grid calculation is data distribution. In most cases, mass spectra files are much smaller than the protein databases they are matched against. The December 2006 release of TrEMBL [18] is about 1.3 GB in size. Distributing the entire file on each target machine can cause data transport and local storage issues. Additionally, [13] demonstrate a log-log-linear reduction of per-spectrum analysis time with increasing size of the set of spectra, whereas the influence of the protein database size on calculation time is proportional. As a consequence, it is opportune to explore the impact of splitting spectrum sets, protein sequence files, or both. The implemented parallelization scheme is shown in Figure 1a. To compensate for the effect of reduced peptide list size on peptide e-values, the output threshold of the non-refined step is lowered proportional to the number of protein database subdivisions, leading to an almost stable number of candidate proteins.

1.3. Grid-Enabled Implementation

End users should not be forced to know all the complexities and peculiarities involved in executing large proteomics analysis jobs in a grid environment. In addition, the use of PVM or MPI is not appropriate for grids due to network latency and node persistence issues. To this end, we have developed a three-layered application service to interface between users and the actual computation (Figure 1b). The service takes care of portal and workflow aspects for proteomics searches and uses existing grid middleware or local resource management systems (LRMS) to submit independent jobs.

The central part of the application service consists of a Perl service daemon, which parallelizes incoming search requests and preprocesses input data. Data sets are deployed to the grid, and jobs are monitored, triggering intermediate or post-processing steps.

The application service interacts with compute resources through a separate layer, which abstracts interaction with the grid middleware or LRMS. This layer has been implemented as a Perl local resource management system interface (LRMSI) module that hides the implementation differences of different grid middleware or batch systems and presents a simple API for handling jobs and job data. This approach has the advantage that changes in middleware versions or even switching from a grid infrastructure to a cluster environment can take place rapidly. The LRMSI is a derivative of the ProtoGRID developed in the context of the SwissBioGrid [19], and already supports a number of common LRMS software.

Requests for analysis are submitted by users through a web portal, and the results of finished analyses can be retrieved and visualized within the portal. We are using a customized version of the Swiss-Model Workspace web application [20] as framework for user-based project and data handling.

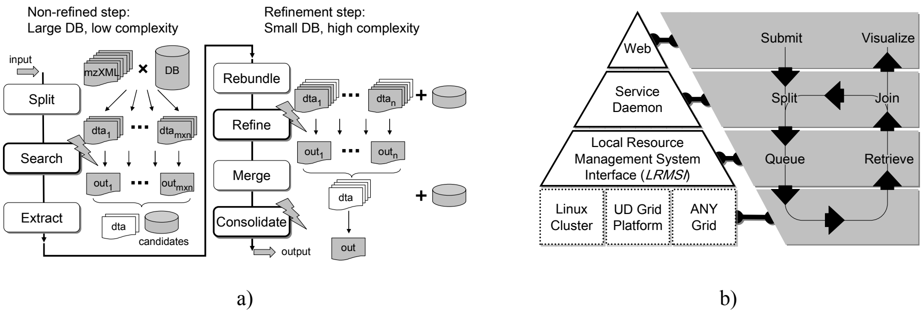


Fig. 1: a) Parallelization scheme of the algorithm. In the non-refined step, work units are formed by the cross-product of *n* sets of spectra and *m* protein database subdivisions. Collected candidate proteins are subjected to a refined search against the original spectra parts (including previously matched spectra) before consolidation. Steps marked with a bolt are executed on the grid. b) Three-layer architecture of the proteomics application service. Requests are received through a web portal. A daemon implements the parallelization scheme and interacts with the underlying compute infrastructure through a LRMS interface layer.

1.4. Validation of Parallelization Scheme

To assess the validity of the proposed grid parallelization scheme, a test case was set up by defining a benchmark consisting of a set of spectra, a standard protein database and common model parameters. Three runs on a single-CPU computer using different analysis protocols (*R1*, *R2*, *R3*) serve as reference to assess the results obtained on the grid. The MS/MS dataset used is a standard set created by analyzing a mixture of 17 proteins from different species on a Micromass MALDI Q-TOF device, provided in mzXML format [21] and obtained from [22]. UniProtKB/Swiss-Prot version 9.1 was used as protein sequence database, containing 241’365 protein sequences (107 MB) [18]. For reference runs and grid analyses, the same x!tandem binary (version 2006-09-15-3) including the k-score plug-in [23] was used running on Microsoft Windows XP.

The peptide expansion model included tryptic digestion with a maximum of 3 missed cleavage sites. Cysteine residue mass was modified by +57 Da to account for cysteine carboxamidomethylation during protein sample preparation. Additionally, the following frequently observed potential modifications were added: methionine oxidation (+16Da), asparagine/glutamine deamidation (+1Da), as well as serine/threonine phosphorylation (+80Da) in refinement.

We define the *R1* reference as the results of a local, single-CPU run of x!tandem. In the *R1* protocol, the whole MS/MS dataset of the mzXML file is matched against the complete native protein sequence database. As the protocol for grid-based analysis comprises more processing steps, several sources of noise can interfere with the final output. In order to get fine-grained information about where score variations may be caused, two modified local runs were carried out that approximate the grid protocol. The *R2* reference was defined as the result of a local standalone run, followed by a consolidation step against the extracted candidate protein list. In this step, previous x!tandem output (including processed spectra and model parameters) is used as input. Here, both spectra set and protein sequences have changed, and the spectra might have undergone minor conversion modifications. A third reference, the *R3* standard, is obtained by converting the MS/MS dataset from mzXML to plain peak lists (DTA format) before executing the R2 protocol. This conversion is a prerequisite for spectra rebundling in the grid environment.

2. Results and Discussion

Peptide and protein e-values both depend on the set of spectra analyzed, the protein database, and on model parameters causing database expansion. Changes in one or more of these factors may strongly influence the resulting e-values both for peptides and proteins. Instability might be increased in protein scores, as in addition to cumulated peptide score shifts and fluctuations, further distortion is injected by the dataset and protein-database related terms in the protein scoring function. To study these effects, we first investigated differences in the results of the three reference protocols, and subsequently compared these to results of the grid-enabled version.

2.1. Reference Performance

The results of both the *R2* and *R3* reference were compared to the output of the single-run *R1* reference analysis. The correlations to the e-values of *R1* are $c_s = 0.94$ for both *R2* and *R3*.

The scatter plot of consensus hits (spectra and proteins that were matched in both analyses) of *R1* and *R2* shows significant fluctuations between corresponding e-values on peptide level (Figure 2). No significant difference was observed between *R2* and *R3*, as spectrum conversion has more influence on the composition of the scored set of spectra than on the e-values of consensus spectra. This means that a considerable amount of *statistical flickering* is injected in the consolidation step, where e-values are based on conditions much different from the original setting. The least-squares linear fit reveals a systematic –0.78 shift of the consolidated log(e-values), corresponding to e-values being almost 6 times smaller in *R2* than in *R1*.

At protein level, e-values show a much higher correlation of $c_p = 0.99$. This indicates that underlying peptide e-value fluctuations are random enough to be compensated in the protein scoring function. However, the peptide e-value shift is cumulated in protein scores and amplified in proteins featuring multiple peptide matches, resulting in increased deviations for highly significant protein identifications.

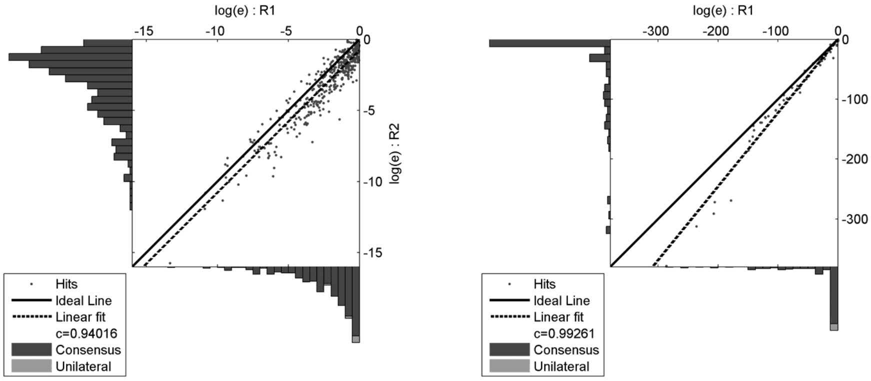
To investigate the impact of the statistical instabilities on the output characteristics, ROC-like sensitivity-selectivity curves of *R1* and *R2* analyses have been plotted in Figure 3. For increasing e-value cutoff levels (decreasing match significance), the number of true positive hits, i.e. matches against peptide sequences present in the known protein mixture, is plotted against the false positives. At spectrum level, no significant difference between the characteristics appears in the first 200 assigned spectra. Beyond, the curves diverge as *R1* yields a few more true positives.

At the level of inferred proteins, there is no significant divergence between the reference schemes. The generally high false positive rate is due to shadow matches by homologous proteins, e.g. from different species.

2.2. Stability of Grid Results

2.2.1. e-Value Correlation

To investigate the stability of the e-value statistics under grid parallelization conditions, the results of grid analyses were compared to the *R1* and *R2* reference e-values. For each grid result, the subset of consensus spectra and proteins were extracted. The distributions of the log(e)-differences (residues) are shown in Figure 4.

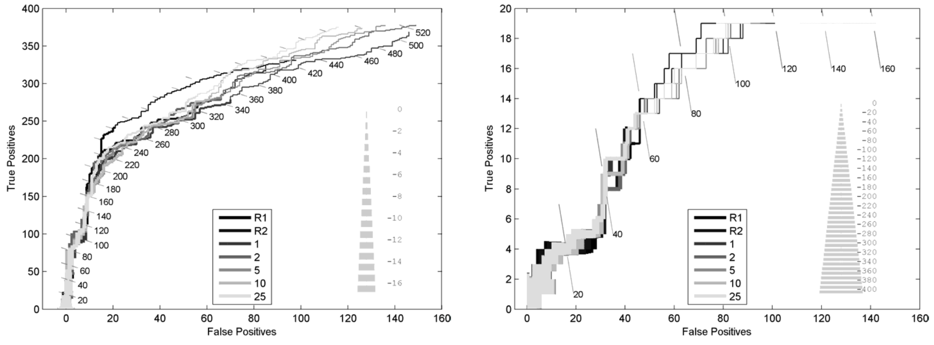


a) b)

Fig. 2: Peptide (a) and protein (b) score correlations of *R2* against the *R1* reference. The histograms show the respective score distributions of consensus spectra/proteins (assigned in both references) (dark) and matches from only one analysis (light).

Varying the number of spectra subdivisions has no significant impact on either spectra or protein score residue distributions. When using different numbers of protein database subdivisions, minor differences in the distributions appear.

Compared to their *R1* counterparts, grid spectra log(e-values) are shifted by approx. –0.8. The negative outliers of protein e-value residues reflect the cumulated shift in high ranking proteins. Compared to *R2*, this shift entirely disappears both for peptides and proteins, confirming the consolidation step as the cause of this shift. Residues spread slightly wider for higher numbers of database subdivisions, as illustrated by both the outliers and the increased inter-quartile distance. This is due to an overcorrection of the adapted output e-value threshold at the non-refined step: about 15% less candidates are selected with 25 protein database subdivisions than when the database is not split. In this way, low-quality matches are removed which score very close to the threshold and reduce residue dispersion.



a) b)

Fig. 3: Spectra (a) and protein (b) ROC-plots of the *R1*/*R2* references and of grid-computed results differing in the number of protein database subdivisions. Line thickness illustrates log(e-value) threshold and the diagonal lines indicate the number of selected matches. The high number of false positives in the protein charts is due to the shadow matches of homologous proteins (e.g. from different species).

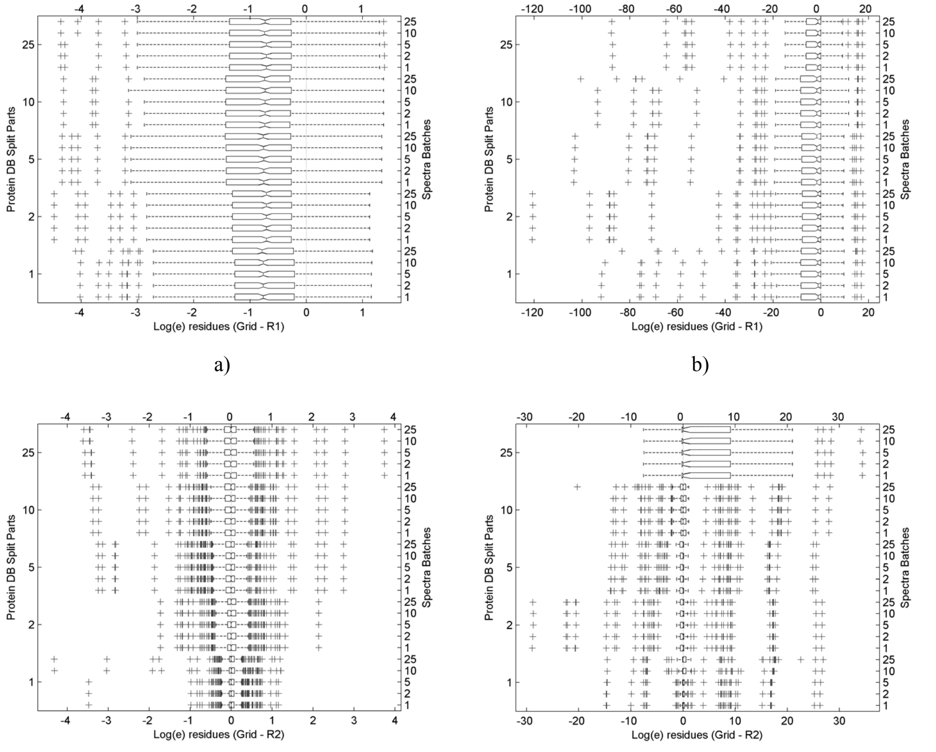


Fig. 4: Residues of spectra and protein log(e-values) between different grid analysis configurations and *R1* (a+b) or *R2* (c+d).

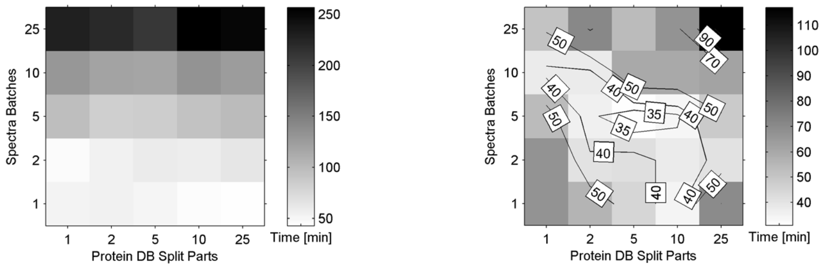
In summary, correlation coefficients of the consensus e-values against *R1* fall in the intervals [0.9327 0.9460] for spectra and [0.9838 0.9880] for proteins. Against *R2*, they improve to [0.9722 0.9972] and [0.9955 0.9974], respectively. Within these bounds, correlations tend to degrade slightly with increasing number of protein database subdivisions.

2.2.2. Detection Characteristics

To compare the analytic power of grid results with local runs, the ROC-curves of different protein database distribution schemes are shown in Figure 3a and b. In this context, the number of sets of spectra plays a minor role and can be neglected.

Grid analyses consistently yielded more spectra hits. However, their true/false-positives ratio is very similar to that of the *R2* reference. Characteristics are almost equal for the best 200 spectra, and differ only in the low-quality part. At protein level, no significant differences appear at all.

These findings, taken together with the e-value correlation data, indicate that the quality of grid results is nearly indistinguishable from searches using the *R2* or the *R1* protocol locally, in particular when considering the top-scoring spectra and proteins, which typically are the most relevant in experiments.



a) b)

Fig. 5: CPU (a) and project wall clock time (b) for different combinations of spectra set and protein database divisions. The contour lines in the wall time chart emphasize the central region of optimal configuration.

2.3. Numerical Stability and Performance Metrics of Grid-Enabled Executable

CPU time depends on the collective CPU power of the machines a job is running on, a quantity that fluctuates in a heterogeneous, dynamic grid resource. We have measured calculation time for 10 identical distributed x!tandem analyses to quantify the magnitude of these variations. Job splitting parameters were set to a batch size of 500 spectra and 10 protein database subdivisions, giving rise to 50 non-refined and 5 refined work units. The Grid MP middleware estimates the amount of CPU time used by a job based on the share of CPU assigned to a particular work unit, integrated over time and all work units of a job. Despite the wide heterogeneity of PC-grid machines, relative standard deviation between the runs was 5.3%. Moreover, all 10 searches resulted in identical output, demonstrating numerical stability of the grid-enabled application.

Next, CPU times were recorded for different schemes of spectra and protein database splitting. Results are shown in Figure 5a. As expected, splitting generally increases the cumulative CPU time, due to multiplication of the program overhead. This effect is particularly prominent when splitting spectra datasets, as the protein expansion step is computationally expensive.

From the user perspective, the total wall clock time for job execution is perceived as more important, defined as the time between job submission and result retrieval, including queuing and transmission overhead. Corresponding measures have been sampled for the same distribution schemes as above (Figure 5b). We identified a valley of optimal configuration in the wall time charts. Performance gain is most pronounced when splitting the protein database into 5 subdivisions speeds up the non-refined step, and when splitting spectra into 5 subdivisions enhances refinement step performance.

3. Conclusions

In the present work, we have demonstrated how MS/MS protein identification can be transformed into a data-parallel task suitable for efficient grid computing. By developing a multi-layer application service, we manage to abstract the parallelization and grid submission process from the user while maintaining an open architecture supporting various LRMS. We have validated our approach by comparing results obtained on the grid with a number of reference result sets that were calculated on a local single-CPU resource.

Although grid processing introduces fluctuations into peptide and protein scores, the resulting peptide and protein score characteristics do not degrade. However, the fact that match selection ultimately relies on e-value calculations that show extreme dependence on boundary conditions inherently causes statistical instability. Here, we demonstrate how this instability can be minimized, but using the present scoring scheme, it can not fully be eliminated. One future direction in this regard is the inclusion of complementary information and descriptors in the process of peptide and protein inference.

We next investigated the best scheme for job parallelization. While an optimum is expected to exist, the exact position depends mainly on 3 job-specific factors: spectra dataset size, protein database size, and protein expansion model complexity. Ideally, parallel submission to many machines decreases overall time, but the computation/data transfer ratio must be kept high to avoid accruing overhead. In bigger experiments (more spectra, larger databases, more complex model), the optimum is likely to shift towards a higher number of database subdivisions. Grid-based optimization strategies, such as the one presented here, are crucial to address the computational needs of large-scale proteomics studies.

Acknowledgments

We thank Paul Jenö, Suzette Moes and Konstantin Arnold for valuable discussions and technical support. **Author contributions:** MP, MM, RA, MCP and TS initiated the project, MP and TS designed the study. DZ and MP developed the application service. DZ designed the experiments and analyzed the data. MP and DZ wrote the paper.

References

[1] S. D. Patterson and R. H. Aebersold, Nat Genet, vol. 33 Suppl, pp. 311-23, 2003.

[2] J. C. Smith and D. Figeys, Mol Biosyst, vol. 2, pp. 364-70, 2006.

[3] M. R. Flory and R. Aebersold, Prog Cell Cycle Res, vol. 5, pp. 167-71, 2003.

[4] T. Sundsten, M. Eberhardson, M. Goransson, and P. Bergsten, Proteome Sci, vol. 4, pp. 22, 2006.

[5] C. R. Cho, M. Labow, M. Reinhardt, et al., Curr Opin Chem Biol, vol. 10, pp. 294-302, 2006.

[6] S. V. Parikh and J. A. de Lemos, Am J Med Sci, vol. 332, pp. 186-97, 2006.

[7] S. Ciordia, V. de Los Rios, and J. P. Albar, Clin Transl Oncol, vol. 8, pp. 566-80, 2006.

[8] S. Schaub, J. A. Wilkins, D. Rush, and P. Nickerson, Expert Rev Proteomics, vol. 3, pp. 497-509, 2006.

[9] A. Schmidt and R. Aebersold, Genome Biol, vol. 7, pp. 242, 2006.

[10] P. Hernandez, M. Muller, and R. D. Appel, Mass Spectrom Rev, vol. 25, pp. 235-54, 2006.

[11] E. A. Kapp, F. Schutz, L. M. Connolly, et al., Proteomics, vol. 5, pp. 3475-90, 2005.

[12] M. P. Washburn, D. Wolters, and J. R. Yates, 3rd, Nat Biotechnol, vol. 19, pp. 242-7, 2001.

[13] R. Craig and R. C. Beavis, Rapid Commun Mass Spectrom, vol. 17, pp. 2310-6, 2003.

[14] D. T. Duncan, R. Craig, and A. J. Link, J Proteome Res, vol. 4, pp. 1842-7, 2005.

[15] R. Craig and R. C. Beavis, Bioinformatics, vol. 20, pp. 1466-7, 2004.

[16] D. Fenyo and R. C. Beavis, Anal Chem, vol. 75, pp. 768-74, 2003.

[17] J. Eriksson and D. Fenyo, J Proteome Res, vol. 3, pp. 32-6, 2004.

[18] C. H. Wu, R. Apweiler, A. Bairoch, et al., Nucleic Acids Res, vol. 34, pp. D187-91, 2006.

[19] M. Podvinec, S. Maffioletti, P. Kunszt, et al., presented at 2nd IEEE International Conference on e-Science and Grid Computing (e-Science 2006), Amsterdam, The Netherlands, 2006.

[20] K. Arnold, L. Bordoli, J. Kopp, and T. Schwede, Bioinformatics, vol. 22, pp. 195-201, 2006.

[21] P. G. Pedrioli, J. K. Eng, R. Hubley, et al., Nat Biotechnol, vol. 22, pp. 1459-66, 2004.

[22] http://sashimi.sourceforge.net/repository.html, Dec 18, 2006.

[23] B. MacLean, J. K. Eng, R. C. Beavis, and M. McIntosh, Bioinformatics., vol. 22, pp. 2830-2, 2006.