BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 22 2006, pages 2830–2832
doi:10.1093/bioinformatics/btl379

Data and text mining

General framework for developing and evaluating database scoring algorithms using the TANDEM search engine

Brendan MacLean[1,2], Jimmy K. Eng[1], Ronald C. Beavis[3,4] and Martin McIntosh[1,*]

[1]Fred Hutchinson Cancer Research Center, Seattle, USA, [2]LabKey Software, LLC, Seattle, USA, [3]Beavis Informatics Ltd, Winnipeg, Canada and [4]University of British Columbia, Vancouver, Canada

Received on May 28, 2006; revised and accepted on July 5, 2006
Advance Access publication July 28, 2006
Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Tandem mass spectrometry (MS/MS) identifies protein sequences using database search engines, at the core of which is a score that measures the similarity between peptide MS/MS spectra and a protein sequence database. The TANDEM application was developed as a freely available database search engine for the proteomics research community. To extend TANDEM as a platform for further research on developing improved database scoring methods, we modified the software to allow users to redefine the scoring function and replace the native TANDEM scoring function while leaving the remaining core application intact. Redefinition is performed at run time so multiple scoring functions are available to be selected and applied from a single search engine binary. We introduce the implementation of the pluggable scoring algorithm and also provide implementations of two TANDEM compatible scoring functions, one previously described scoring function compatible with PeptideProphet and one very simple scoring function that quantitative researchers may use to begin their development. This extension builds on the open-source TANDEM project and will facilitate research into and dissemination of novel algorithms for matching MS/MS spectra to peptide sequences. The pluggable scoring schema is also compatible with related search applications P3 and Hunter, which are part of the X! suite of database matching algorithms. The pluggable scores and the X! suite of applications are all written in C++.

Contact: mmcintosh@fhcrc.org

Availability: Source code for the scoring functions is available from http://proteomics.fhcrc.org

Supplementary information: http://proteomics.fhcrc.org

1 INTRODUCTION

A key component in the rapid growth of proteomic research has been the development of algorithms to sequence peptides and proteins using a database scoring algorithm that compares tandem mass spectra (MS/MS) of peptides against sequences in protein databases. In spite of their central role in peptide sequencing, implementations of these algorithms have historically been proprietary in nature and distributed by instrument vendors or software companies as part of larger and more comprehensive software platforms. In addition, there has been little activity by the computational research community on developing frameworks for testing and implementing better scoring algorithms. More recently, MS/MS database search tools such as TANDEM (Craig and Beavis, 2004), OMSSA (Geer *et al*., 2004), and ProbID (Zhang *et al*., 2002) have been made open source and freely available to the proteomics research community. The framework for various MS/MS database search tools are all conceptually very similar, including components for reading in input spectra, parsing sequence databases, identifying candidate peptides of the same mass as the input spectra, comparing sequence versus spectrum to generate a score, storing relevant peptide hits as the database is being searched, and lastly writing out the search results. Re-implementation of a new score function has historically required the development of an entire new application.

To develop a platform for implementing and testing novel score functions we have extended the TANDEM application to contain a ‘pluggable’ peptide scoring function that allows users to redefine the scoring function while leaving the remaining TANDEM application intact, including its support for data input, generating candidate peptides to score, complex refinement mode queries, assembling protein level summaries and cross platform compatibility. The development of a pluggable scoring framework is intended to foster research in the area of MS/MS database searching by allowing computational scientists to focus on developing more sensitive and discriminating score functions, arguably the key element differentiating various MS/MS search tools, and easily deploy them to the proteomics community in a high-throughput application.

2 METHODS

A key requirement for our implementation of flexible pluggable scoring was to allow changes to the scoring algorithm without any need to modify the distributed source code. We first placed much of the existing TANDEM scoring class (mscore) into a base class for scoring plug-ins. In that base class, we separated the code that provides general scoring infrastructure from the code that specifies scoring choices made by the native TANDEM scoring algorithm. Infrastructure for scoring functions, such as a state machine for choosing which peptide sequences to score, was left in the base class. Specific scoring decisions were moved into functions that were then made virtual so they could be overridden by other plug-ins. These virtual functions, which represent the true pluggable scoring application programming interface (API) are marked within the mscore.h file in the TANDEM project. A specific scoring function is selected at run time by specifying a search parameter; this

[*]To whom correspondence should be addressed.

enables a single installation to support multiple scoring functions simultaneously within a single search engine. A plug-in manager is provided that maps a plug-in type to a set of named plug-ins (see mplugin.h and mplugin.cpp in source distribution). Scoring plug-ins are each implemented in a single C++ header file (.h) and source file (.cpp). They can be added or removed by adding or removing those files from the project and recompiling; no code in the other files or the Makefile needs to be modified. It can be compiled on most systems with a C++ compiler.

The scoring plug-in has access to spectrum information, including parent mass, charge and intensity, measured ion masses and intensities, as well as sequence informations, including sequence mass, length and amino acid string, and ion masses. For a complete list, see spectrum.h and mscore.h.

2.1 Scoring function implementations

The following scoring functions use this scoring API and are currently freely available either as part of the TANDEM project or from http://proteomics.fhcrc.org. Further considerations for implementing new scoring functions are provided in Supplementary Material.

TANDEM/Native. The algorithmic decisions from native TANDEM is contained in a derived class called 'mscore_tandem'. In approximate terms, the native score, based on the hypergeometric distribution, is calculated as the sum of matched peak intensities multiplied by the factorial of the number of matched b-ions and the factorial of the number of matched y-ions (Craig and Beavis, 2004).

TANDEM/*K-score*. The class 'mscore_k' described by Keller *et al*. (2005) is based on summing the matched peak values of input spectra following a pre-processing step intended to give a more sensitive match by taking contributions from noise and unmatched peaks. This is contrasted with simply summing matched peak intensities from an unprocessed spectrum which would only take contributions from matched peaks. This score function is also compatible with PeptideProphet (Keller *et al*., 2002).

TANDEM/*S-score*. This simple score, contained in a class 'mscore_s', is a very basic scoring function from which researchers can begin developing their own scoring functions. This score sums the log intensities of matched peaks, then divides by the square root of the sequence length. Log rather than raw intensities are used in order to reduce the effect of highly dominant peaks and the sequence length adjustment compensates for the higher rates of random matches that may occur with peptides having larger numbers of amino acids.

3 RESULTS

We demonstrate the pluggable TANDEM search engines using a publicly available yeast dataset obtained from the public PeptideAtlas repository (Desiere *et al*., 2005). The 'Comp12vs12standSCX' yeast dataset (freely available at peptideatlas.org/repository) is a non-contrived, complex protein sample acquired on an ion trap instrument. The Native, *S*-score and *K*-score function plug-in modules were all compiled into the same executable binary. Searches were performed against a database composed of 5873 yeast ORF sequences from SGD (Hong *et al*., ftp://ftp.yeastgenome.org/yeast/) appended with 188 752 reversed sequences from UniProt (Apweiler *et al*., 2004). We will use these data to demonstrate the scoring modules by counting and comparing the number of identifications that match with the forward database (correct matches) compared with the reverse database (false positives).

We apply the TANDEM application to this dataset while choosing the score function at run time. We summarize the accuracy of scores by sorting the scores from highest to lowest, choosing a number (n) of top scoring peptides, and then computing the number of correct (matching yeast database) and incorrect peptides (reverse database) among them, denoted Y(n) and X(n), respectively. We characterize the overall accuracy of each scoring algorithm by plotting Y(n) versus X (n) for all n. The intent of this application note is to establish the feasibility of the pluggable scoring framework, not to compare the individual algorithms. The curves in Figure 1 plot the X(n) versus Y(n) for the yeast dataset and demonstrate rough performance equivalence between the Native and *K*-score functions and, as expected, poorer performance using the simpler *S*-score plug-in.

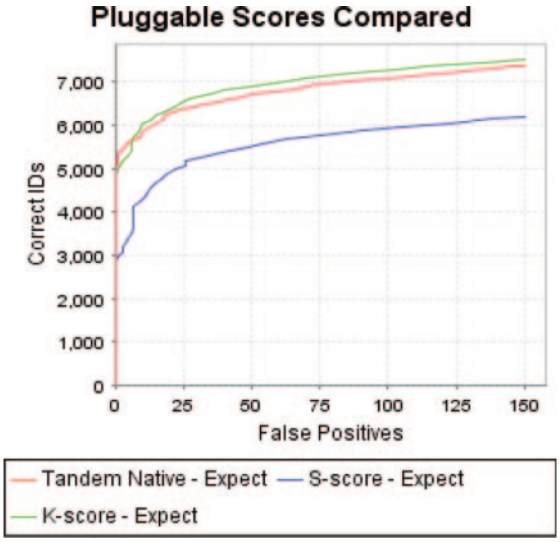

Fig. 1. ROC curves for Native, *K*-score and *S*-score functions' performance on the yeast dataset.

4 CONCLUSION

We have extended TANDEM to easily incorporate new pluggable score functions. We demonstrate two new implementations in addition to the Native score function: *K*-score, which implements a previously published algorithm, and *S*-score, which provides a simple template algorithm that researchers can start with when beginning research on new methods. We do not recommend using the *S*-score scoring algorithm in production analysis, but only as a starting point for further research. These data and searches are used here to demonstrate the pluggable scoring framework and offer base dataset where researchers compare new methods. We did not comprehensively compare the scoring algorithms, which is outside the scope of this application note. This extension builds on and is a demonstration of the success of the open-source TANDEM project, and it should facilitate advances in research on more advanced algorithms for protein identifications.

ACKNOWLEDGEMENTS

This work was funded by National Cancer Institute contract 23XS144A. Funding to pay the Open Access publication charges was provided by National Cancer Institute Subcontract 23XS144A.

Conflict of Interest: none declared.

REFERENCES

Apweiler,R. *et al*. (2004) UniProt: the Universal Protein knowledgebase. *Nucleic Acids Res*., **32**, D115–D119.

Craig,R. and Beavis,R.C. (2004) TANDEM: matching proteins with tandem mass spectra. *Bioinformatics*, **20**, 1466–1467.

Desiere,F. *et al*. (2005) Integration with the human genome of peptide sequences obtained by high-throughput mass spectrometry. *Genome Biol*., **6**, R9.

Geer,L.Y. *et al*. (2004) Open mass spectrometry search algorithm. *J*. *Proteome Res*., **3**, 958–964.

Hong,E. *et al*. Saccharomyces Genome Database.

Keller,A. *et al*. (2002) Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal*. *Chem*., **74**, 5383–5392.

Keller,A. *et al*. (2005) A uniform proteomics MS/MS analysis platform utilizing open XML file formats. *Mol*. *Syst*. *Biol*., EPub.

Zhang,N. *et al*. (2002) ProbID: a probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. *Proteomics*, **2**, 1406–1412.