

Research article

Peptizer: a tool for assessing false positive peptide identifications and manually validating selected results

Kenny Helsens^{1,2}, Evy Timmerman^{1,2}, Joël Vandekerckhove^{1,2}, Kris Gevaert^{1,2}, Lennart Martens^{1,2,3}

¹Department of Medical Protein Research, VIB, B-9000 Ghent, Belgium

²Department of Biochemistry, Ghent University, B-9000 Ghent, Belgium

³EMBL Outstation, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK

Running title: Peptizer

Corresponding author: Prof. Dr. Kris Gevaert, Department of Biochemistry, Faculty of Medicine and Health Sciences, Ghent University, A. Baertsoenkaai 3, B-9000 Ghent, Belgium. Tel.: +32-92649274; Fax: +32-92649496; E-mail: kris.gevaert@ugent.be.

E-mail addresses: KH: kenny.helsens@ugent.be; ET: evy.timmerman@ugent.be; JV: joel.vandekerckhove@ugent.be; KG: kris.gevaert@ugent.be; LM: lennart.martens@ebi.ac.uk

Abbreviations:

COFRADIC: Combined FRActional DIagonal Chromatography

SCX: strong cation exchange

TNBS: 2,4,6-trinitrobenzenesulfonic acid

GUI: Graphical User Interface

Summary

False positive peptide identifications are a major concern in the field of peptide-centric, mass spectrometry driven gel-free proteomics. They occur in regions where the score distributions of true positives and true negatives overlap. Removal of these false positive identifications necessarily involves a trade-off between sensitivity and specificity. Existing post-processing tools typically rely on a fixed or semi-fixed set of assumptions in their attempts to optimize both the sensitivity and the specificity of peptide and protein identification using MS/MS spectra. Because of the expanding diversity in available proteomics technologies however, these post-processing tools are often struggling to adapt to emerging technology-specific peculiarity. Here we present a novel tool named Peptizer that solves this adaptability issue by making use of pluggable assumptions. This research-oriented post-processing tool also includes a graphical user interface to perform efficient manual validation of suspect identifications for optimal sensitivity recovery. Peptizer is open source software under the Apache2 license and is written in Java (<http://genesis.ugent.be/peptizer>).

Introduction

The protein set of a biological system is the topic of research in proteomics, with bottom-up proteomics approaches relying on peptides as the fundamental analytical unit. Typically, proteins are extracted prior to being digested into peptides, generally by a specific protease such as trypsin. In most workflows, the highly complex peptide sample obtained after digestion is then separated in one or more chromatographic dimensions before being analysed by a mass spectrometer. Peptides are ionized and fragmented in this instrument, yielding fragment ion spectra as the final experimental output **(1)**. Data interpretation algorithms are then employed to identify the peptide of origin from the fragment ion spectrum. The final step in the identification procedure consists of assembling a protein list from the identified peptides **(2)**.

As a first and crucial step of data interpretation, coupling of a fragment ion spectrum to a peptide sequence has attracted much effort aimed at optimizing this process. A recent review of the variety of methods and tools available for this purpose can be found in **(3)**. The most commonly applied method is based on sequence database searching by database search engines such as SEQUEST **(4)**, Mascot **(5)**, X!Tandem **(6)**, VEMS **(7)** or OMSSA **(8)**. The overall concept behind these algorithms is similar and consists of the generation of theoretical fragment ion spectra from sequence database entries, against which experimental fragment ion spectra are matched. The difference between the algorithms is usually found in the spectral comparison method and scoring scheme **(9)**. The most difficult part of this analysis is not necessarily finding the best match from the sequence database, but finding out whether this best match is actually valid. Indeed, an experimental spectrum can not always be compared to the actual theoretical spectrum of its original precursor, as this precursor may be absent from

the database, or because the precursor peptide carried one or more unanticipated modifications. Even so, this experimental spectrum may still be matched with a considerable score to a theoretical fragmentation spectrum derived from an unrelated precursor. In order to filter out such background matches, several search engines include probability-based scoring algorithms **(9)**, in which the score of a proposed peptide identification can be compared against a threshold score for a given confidence level. In addition, post-processing tools have been developed that analyze the detailed output of a search engine in order to obtain a revised score that should further optimize sensitivity and specificity **(10-13)**. Typically, such algorithms rely on certain assumptions about the identifications to model true positive and true negative score distributions. PeptideProphet **(13)** for example relies on a mixture model approach that models SEQUEST score distributions according to fixed assumptions such as tryptic correctness of the identified peptides. Ultimately, a revised probabilistic score is calculated that should allow discrimination between true and false positives with increased accuracy.

In certain cases however, only a subset of all peptide identifications obtained are of relevance to the biological system under study. In these cases, expert manual validation of the identifications is a more commonplace strategy for quality control. Protein modification studies for example, often find biological relevance in a small subset of all experimentally obtained data **(14, 15)**. In addition, the so-called "single hit wonders", which often populate the majority of identified peptides or proteins in gel-free proteomics, should not be simply discarded but must be treated intelligently as they potentially contain valuable biological information **(16, 17)**. The manual validation required to assure the reliability of the biological conclusions drawn from such peptide identifications can be performed by using the visualization tools included with the search engine, or by specialized applications such as CHOMPER **(18)**, DTASelect **(19)**, or

MyProMS **(20)**. These tools present a specific set of details on a peptide identification and its associated spectrum for user validation. Finally, a semi-manual option was recently added to PeptideProphet by allowing the user to enable or disable certain of the modelling assumptions from which the overall score is derived **(21)**.

An important side effect of the evolution of proteomics technologies towards more specialized and targeted approaches **(22, 23)** however, relates to the corresponding changes in the actual assumptions that can be made about the identifications. These changes effectively introduce new parameters that can be used to further enhance the separation of false and true positives, yet are necessarily largely ignored by tools built upon fixed, generalized assumptions. In order to allow this expanding array of technologies and associated identification parameters to be used effectively in the post-processing and validation of proteomics data, we here present the Peptizer tool. Built upon a dynamic profiling framework that operates on pluggable assumptions, Peptizer can be quickly and efficiently configured with any *a priori* knowledge that is available to the user. Each assumption or parameter is coded in an autonomous agent, which is allowed to cast a vote on each peptide identification. In a second layer, the votes of these agents are aggregated using a pluggable algorithm, which outputs a final score that is used to judge whether an identification represents a potential false positive. We show that elimination of these suspicious identifications increases specificity, albeit at the cost of a noticeable loss in sensitivity through removal of certain true positives. A sophisticated and highly efficient manual validation interface is also included which can be used to compensate in part for this loss in sensitivity.

Experimental Procedures

MS/MS data

The MS/MS spectra used in this study have been published previously **(24)**. Full experimental details are provided in the Supplementary information that can be downloaded from the journal's website. Briefly, human K562 cells were lysed by cycles of freeze-thawing, followed by reduction and alkylation of cysteines. Primary free amines were then trideutero-acetylated by N-hydroxysuccinimide trideutero-acetate. Alkylated and acetylated proteins were digested by trypsin and the generated peptide mixture was separated by strong cation exchange at pH=3 to enrich for α -amino-blocked peptides in the SCX non-binding fraction. The sample was then acidified to oxidize methionines before the primary N-terminal COFRADIC separation **(25)**. Fractions of 4 min wide were collected and treated with TNBS. Such modified primary fractions were then loaded for the secondary COFRADIC run wherein the α -amino-blocked peptides – which show no altered chromatographic properties – are collected. The secondary fractions were analyzed by LC-MS/MS using a microfluidic interface (Agilent's Chip Cube) on an Agilent XCT-Ultra ion trap mass spectrometer operated as described in **(26)**.

Peptide identification, false positive estimations and Peptizer development

The MS/MS spectra were searched by Mascot version 2.2 against the human subset of the UniProtKB/Swiss-Prot sequence database, release 53.2 (June 26, 2007), concatenated with a shuffled version of this database generated by DBToolkit **(27)**. The following parameters were used in the Mascot searches: peptide mass tolerance and peptide fragment tolerance were set at ± 0.5 Da, and allowed precursor charges were set to 1+, 2+, and 3+. Fixed modifications were oxidation of methionine to its sulfoxide derivative, trideutero-acetylation of lysine and

carbamidomethylation of cysteine. Pyroglutamate formation (N-terminal Gln), pyrocarbamidomethyl cysteine formation (N-terminal carbamidomethylated cysteines), acetylation and trideutero-acetylation of the α -N-terminus and deamidation (Gln and Asn) were considered as variable modifications. Endoproteinase Arg-C/P was set as the proteolytic enzyme and at most 1 missed cleavage was allowed. Mascot's instrument setting parameter was set to ESI-TRAP. Only MS/MS-spectra receiving an ion score equal to or exceeding Mascot's identity threshold score at the 95% confidence level were withheld for further inspection by Peptizer. All experimental fragmentation spectra (32,403), peptide identifications (2,739) made in the 'forward' protein database and corresponding experimental details are made publicly available via the PRIDE database **(28)** at <http://www.ebi.ac.uk/pride/> under the experiment accession number 3261. However currently these data can only be assessed by referees of the manuscript (login: review13652, password: NsxEqpGm).

To estimate the false positive distribution we performed Mascot searches against a concatenated decoy database as described in **(29)**.

Peptizer was developed as an open source project under the Apache2 license in Java 1.5 (<http://java.sun.com>). Peptizer relies on Mascotdatfile **(30)** to process Mascot result files, and can also interface with the ms_lims software package **(31)**.

Manual validation

Manual validation was performed by an experienced mass spectrometrists. The scientist was blinded to the origin of the peptide identifications (i.e., from the decoy or target set proteins). The scientist was told to apply stringent criteria during the validation. The net effect of the

manual validation was obtained by inspecting the unblinded results after completion of the validation.

Peptizer configuration

Peptizer was configured to employ the agents listed in Table 1 for detecting potential false positive identifications in this dataset. The agent configuration textfile, which can be loaded in Peptizer, can be found at the project website (<http://genesis.ugent.be/peptizer/peptizer/download/profiles.html>). The 'best hit' agent aggregator was used to combine the individual agent votes. The aggregator used simply summed all votes together and marked the peptide identification as suspicious if the result was equal to or greater than two (or when an agent with veto rights declines).

Results

Peptizer was developed as a post-processing tool aimed at separating true and false positive peptide identifications in a highly configurable manner, without relying on any built-in assumptions. Indeed, considerable variations in expected output are often found between distinct research methodologies, that all convey some form of *a priori* knowledge that can ultimately be used to separate identification candidates at the post-processing level. Since existing tools commonly rely on fixed assumptions that are derived from generalized or idealized research methods, they are limited in the amount of *a priori* experimental information they can take into account. In contrast, Peptizer is inherently designed with the necessary flexibility to integrate any available *a priori* knowledge.

Construction of a Peptizer profile

The peptide identifications are tested by evaluating a series of user-selectable and extensible properties. The result of this evaluation can be to decline, reserve or recommend the identification based on that property. The results across all considered properties are then combined in an overall score for identification reliability which can ultimately be filtered upon.

In Peptizer, a property is inspected by an Agent, and the combination of multiple Agent scores is performed by an Aggregator. These two components are shown in **Figure 1** and are discussed in detail in the following sections.

Agents for the inspection of identification properties

An Agent in Peptizer typically inspects a single property of a peptide identification and reports a score (or 'vote') to indicate whether it declines, reserves, or recommends the identification (score of +1, 0, or -1, respectively). An individual Agent can be given veto privilege, which means that a decision to decline an identification by such an Agent will directly result in declining of the identification, irrespective of the votes of the other Agents. Examples of properties that an Agent can inspect include: the peptide sequence coverage by fragment ions, the length of a peptide, the peptide modification status, the difference between peptide ion score and identity threshold, and the difference between best scoring hit and second-best hit, amongst many others.

Furthermore, apart from being readily included in or excluded from a profile, each Agent can be parameterized as well. The Agent that inspects on peptide length, for instance, can be provided with a cut-off length below which to decline an identification. Another example is the Agent that inspects for sequence coverage by b-ions, which also takes a threshold level of coverage below which identifications are declined by the Agent. As a final example, consider the Agent that inspects identifications for missed cleavages; in this case, both the cleavage specificity of the protease as well as the number of tolerated missed cleavages are Agent parameters. Cleavage specificity is therefore easily adapted when evaluating data from an experimental protocol that employs a different protease.

Aggregators for combining Agent votes into an overall score

As outlined above, all peptide identifications are inspected by a voting panel composed of user-selected Agents that each decline, reserve or recommend an identification by casting a vote. These individual votes must then be aggregated into an overall score for the identification, on which recommendation or rejection is ultimately based (see **Figure 1**). A first method in which Agent votes can be combined is by simple summation of the Agent scores. If the end result is above a preset threshold (e.g., 0), the identification is rejected. A more pessimistic approach counts only the number of Agents that decline the peptide identification. If that number is higher than a preset cut-off, the peptide identification is considered bad. Obviously, an Aggregator can also be much more sophisticated than these simple examples, utilizing a support vector machine, neural network or other learning algorithm for instance. Interestingly, Peptizer also supports pluggable Aggregators, thus allowing complete flexibility at both the Agent and Aggregator level. It is worth noting that the Peptizer framework can therefore provide an extremely convenient infrastructure basis for the development and implementation of novel computational strategies for discovering false positive identification profiles.

Availability of Peptizer and providing extensions to the framework

Peptizer is released as open source under the Apache2 software license, and binaries as well as source code can be downloaded from <http://genesis.ugent.be/peptizer>. Even though it is made freely available under a permissive license, the source code is not required to build extensions to Peptizer, nor is a recompilation of the application necessary to include novel Agents or Aggregators. A typical Agent is only about 20 lines of code while a typical, simple Aggregator is about twice that size. Peptizer loads its Agents and Aggregators from a simple XML-based

configuration file upon application start-up, so simply adding a newly developed Agent into this configuration file will make it available for inclusion in the application's voting panel, and the same holds true for Aggregators. The effort required to provide Peptizer with new Agents or Aggregators is thus minimized by design, allowing rapid adoption of novel experimental methodologies and their corresponding *a priori* information through custom-developed Agents and Aggregators.

While Peptizer currently only accepts Mascot '.dat' result files as input, the source of peptide identifications can also be modified. However, in order to extend the reach of Peptizer to other search engine output files, a basic understanding of programming in Java is required as parsing of these more complex files can be more involved. All these extensions to Peptizer can be achieved by implementing well-documented interfaces, thus providing a clean and efficient develop-by-contract approach.

Operation modes of Peptizer and the manual validation interface

Peptizer can be used in one of two modes: fully automatic command-line execution, or semi-automatic operation by means of a user-friendly graphical user interface (GUI). Both modes address a distinct group of users: while the average user will work most comfortably in GUI mode, more experienced users will benefit from the automated and scriptable command-line execution. An important difference between the two modes is that in automatic mode, all suspicious identifications will be considered incorrect, while the GUI mode will simply flag these for further manual validation. The GUI mode thus effectively employs the user as the final arbiter while the command-line mode does not include this final evaluation step.

The Peptizer GUI is designed for optimal efficiency as it guides the user through the process of choosing a data source, creating an Agent profile, and choosing an Aggregator, as shown in **Figure 2**. The top panel takes the source of the peptide identifications, while the center panel is subsequently used to construct the voting panel. Note that Agent parametrization as well as assignment of veto privileges is also taken care of at this stage. The lower panel presents the available Aggregators to the user, and the bottom panel can be used to define the confidence level below which identifications will not even be considered. When the profile configuration is complete, a new task can be started by clicking the appropriate button.

Both user-friendliness and efficiency were optimized by adding extra features in this dialog: tooltips describe the voting logic of an agent, and information on the combination methods of the Aggregators can be called up. More importantly, individual voting panels as well as overall task configurations can be saved for later use. Simple reloading of such a configuration file will reconstruct the exact settings, thus saving the user time while strongly enhancing consistency. These saved configuration files are also readily archived if necessary, can be shared with other researchers, or may serve as preset configurations for command-line execution of Peptizer.

Upon submitting a task, the software starts to analyze each proposed identification using the user-configured Agent profile and Aggregator, and then forwards the results to the manual validation application shown in **Figure 3**. The screen is divided into three major parts: a tree with spectra and identifications on the left, the identification detail view on right and center, and a status panel at the very bottom. Each of these parts can be resized or even collapsed according to the needs of the user. The tree structure fulfils several functions. First, it provides an overview of the work done by colour-coding identifications based on their status (unresolved, user declined, user accepted). Second, it also allows the user to quickly browse the

entire set of suspect identifications. Third, it can be filtered to reveal specific subsets of these suspect identifications. Each tree node holds a single fragmentation spectrum with all of its suggested confident peptide identifications, the number of which is indicated between brackets after the spectrum number. Unfolding the tree node shows these confidently assigned peptide sequences. Applying filters to the tree enhances navigation through the peptide identifications, for example by hiding all identifications that have already been validated. By double clicking a node, a new tab is opened in the detailed view on the right. In this view, three different perspectives are given for the user to explore. Topmost is the annotated modified peptide sequence, consisting of all identified b- and y- ions, annotated as bars on the sequence. The height of the bars indicates the intensity of the corresponding peaks in the spectrum relative to the most intense identified fragment ion. The middle section of the detailed view sports an interactive display of the annotated fragmentation spectrum, while the bottom of the view is taken up by a table. The columns in this table correspond to the significant peptide hits obtained for this spectrum (three in the example given in **Figure 3**), apart from the leftmost column which always serves as a legend. By default, the most confident peptide identification is selected when a new tab is opened, but the user can modify this selection by clicking on another column in the table (column selection is indicated by a darker colour tone). When the selection changes, the experimental fragmentation spectrum shown in the center of the screen is updated with the fragment ion annotations of the selected peptide. Additionally, the annotated sequence in the top panel of the detailed view is also adapted to the newly selected peptide.

Each row in the data table describes a distinct type of general or Agent-derived information. Examples of general information, which is always available regardless of profile composition,

include the peptide sequence, Mascot ion score, Mascot identity threshold, b- and y-ion coverage, precursor mass error, etc. The Agent-derived information obviously depends on the selected Agents in the profile. Typesetting of the individual Agent reports is dependent on the actual vote cast by that Agent for that peptide. For instance, when an Agent that requires the peptide length to be longer than 8 amino acids declines a 7 amino acid long peptide, that row will display "7" in a **bold** typeface, highlighting the fact that this property failed Agent scrutiny. The report is shown in *italics* when the peptide identification is recommended by the Agent. The table therefore functions as a very compact and easily interpretable source of information on the different peptide identifications.

After careful inspection of a spectrum and its peptide identifications, the user may either choose to accept or to reject an identification by clicking on the corresponding buttons in the lower right corner (see **Figure 3**). The red 'STOP' icon rejects an identification, while the green 'OK' icon accepts. Note that accepting one peptide candidate when multiple are given for a spectrum, automatically rejects these other possibilities. For each of these icons, an alternative is given that takes a validation comment to go with the decision (see dialog on **Figure 3**). Once the decision is communicated by a click on the appropriate button, the application will automatically close the freshly validated tab and open up the next available, unresolved identification.

The set of peptide identifications can be saved to the hard drive at any time, and can be reloaded in another session, enabling discontinuous manual validation. Validation data in this form can also be distributed to other users or archived as training material.

The end result of the manual validation can be saved into delimited text files, allowing the user to choose the table data that is included, as well as optionally including the confident peptide identifications that were recommended by the voting panel (and therefore automatically catalogued as good). Moreover, Peptizer can also output its data to a file format that is directly readable by the open source Weka machine learning library **(32)** for further analysis.

Comparison between fully automatic and manual Peptizer validation

Peptizer is a post-processing tool aimed at identifying false positive peptide identifications. False positives have been shown to be simulated by performing decoy searches with experimental spectra **(29)**. By integrating properties of such decoy-derived false positive matches as well as experimental knowledge of mass spectrometry scientists, a Peptizer voting panel was configured to select potential false positive identifications (see **Table 1** which details the agents extracted from the N-terminal COFRADIC dataset reported in **(24)**). A pessimistic Aggregator was chosen and configured to label a peptide identification as suspicious if two or more agents declined the peptide identification.

Because the assignment of an MS/MS spectrum to a peptide sequence is the first and most important step in the identification of proteins, and since the interference from protein inference has not yet been introduced at this level **(2)**, we decided to evaluate Peptizer on results obtained at the level of peptide identification. Improving the quality of peptide identifications will in turn affect protein identification, since more reliable peptides are instrumental in obtaining reliable protein identifications **(33)**.

We assessed the efficacy of the above Peptizer profile in labeling suspicious peptide identifications by applying it to a blinded set of 2,795 peptide identifications obtained by searching a concatenated normal/decoy database. 56 peptide identifications were derived from the decoy database, thus predicting that the whole set of 2,795 peptide identifications is composed of 112 false positives (about 4%) and 2,683 true positive identifications (calculations based on the work of the Gygi lab (29)). The detailed results of applying the Peptizer profile to this dataset are shown in **Table 2**. In total, 193 peptide identifications were labeled suspicious by Peptizer. Among these, 47 peptide identifications originated from decoy sequences and we therefore estimate that this selection contains 83.9% of all false positives (or 94 of the expected 112) in this dataset; a very considerable enrichment.

This set of 193 suspect identifications can then be processed according to two different scenarios. First, full automatic mode can be applied, which simply discards all the suspect identifications. This results in the removal of appr. 83.9% of all false positive identifications (94 identifications), but at the cost of removing appr. 3.7% true positives (99 out of 2,683 identifications) as well. It is worth noting that although the original dataset contained an estimated 4% false positives, it only retained 0.7% false positives (18 out of 2,602 remaining identifications) after applying the Peptizer profile in fully automatic mode.

The loss in sensitivity (removal of 3.7% true positives) in full automatic mode can be partially offset by performing manual validation. Indeed, in this semi-automatic mode the user discarded 158 peptide identifications containing 45 of the identifications made in the decoy database. It is thus estimated that this rejected set of identifications contained 90 false positives and 68 true positives. Since the user also accepted 2 decoy peptide identifications, we could estimate that a total of 31 predicted true positive identifications were accepted by the user, indicating that

about 30% of the true positive identifications rejected by Peptizer in full automatic mode were now “rescued” by the user (**Table 2**). However, the user also made mistakes, albeit of minor influence: 2 of the 47 decoy peptide identifications detected by Peptizer slipped through the user’s scrutiny, representing a negligible increase in total false positives in the final dataset. The time cost for validating these 193 peptide identifications by an experienced user was two working days. Despite all efforts at making the manual validation process as efficient as possible, it does incur a certain time cost. The choice between full automatic or semi-automatic mode must therefore be made based on the importance of sensitivity in the actual experiment. Overall however, usage of Peptizer resulted in a major increase of specificity, at a reasonable cost in sensitivity.

Discussion

Database search algorithms must ultimately rely on fixed assumptions due to their general applicability. Analogous to pathogens that present endogenous material by molecular mimicry, the confluent transition of the scores of true negatives and true positives shows that a database search algorithm sometimes faces a similar problem as the immune system: the good and the bad look very much alike when evaluated by limited, generalized means. The various proteomics approaches however, each contribute new protocol-specific knowledge or assumptions that can be used in the peptide identification sorting process. Since these method-specific validation criteria are not generally applicable, implementation in database search algorithms would intervene with the robustness of that algorithm, on top of being very cumbersome to implement. In order to efficiently make use of this heterogeneous and changing methodology-related information, we here described the implementation and application of Peptizer, a fully configurable post-processing tool that relies on an extremely versatile pluggable voting mechanism.

Ultimately, decisions on sensitivity and specificity typically made by bioinformaticians should match requirements set by experimentalists. As such, the quality of the peptide identifications usually is the highest priority, although specific endeavours such as biomarker discovery will also benefit from maximum sensitivity. The extremely configurable nature of Peptizer readily accommodates these varying circumstances through a custom aggregation of voting results.

It is also important to note that this extreme customizability of Peptizer at various levels is what sets it apart from any other existing tool. A statistical evaluation of Peptizer's validation efficiency compared to other tools was here omitted since determination of the variance specific

of the tools was impractical. However, compared to other semi-manual post-processing applications such as CHOMPER **(18)**, DTASelect **(19)** or MyProMS **(20)**, Peptizer stands out by being fully configurable. While these existing tools may allow the configuration of a fixed set of criteria, Peptizer has no fixed set of criteria. Indeed, Peptizer allows any combination of criteria to be used through its fully configurable and extensible Agent profile. Obviously, as with the existing applications, once an Agent profile is created in Peptizer, each Agent can be configured in detail through parameters. The configurability of Peptizer goes even further however, since even the actual score calculation module can be fully configured by the user through pluggable Aggregators. Importantly, the versatility of Peptizer is functionally connected to both the full automatic and manual modes of operation. Indeed, the information table in the GUI is directly fed by information from the Agents that were selected in the profile, and the nature of each Agent's vote is indicated in the typeface of its detailed report. The manual validation interface thus seamlessly adapts to any user-configured Agent profile, even when it includes custom-written Agents contributed by the user. Full automatic mode supports plugging in advanced, custom-built Aggregators that can connect to machine learning libraries **(32)**.

The agents presented in **Table 1** are mainly based on protein chemistry and peptide identification principles. Even if more or other agents were created, e.g., based on peptide fragmentation patterns and rules extracted from large scale studies on MS/MS-spectra **(34, 35)**, the results presented here show that false positive identifications are already highly enriched in the identifications selected by applying an appropriate Peptizer profile, thus ensuring substantially increased stringency at only limited cost in sensitivity. Furthermore, careful manual validation of the selected subset of peptides using the Peptizer validation GUI

has been shown to maintain specificity, while providing a large sensitivity bonus compared to full automatic processing.

To our knowledge, we also present the first experimental data on the cost of manual validation that is often only hinted upon in manuscripts. In the rich and user-oriented manual validation environment that Peptizer presents, peptide identifications were validated at a rate of about 100 a day. Additionally, instead of having to validate all 2,795 original peptide identifications, only a Peptizer-selected subset of 193 suspicious peptide identifications needed validation. The total cost amounted to two working days validation time, while 80% of the false positives were successfully removed with only 2.5% true positives lost. In the context of a complete proteomics experiment, two days of validation time should be well within acceptable bounds when optimal identification stringency at high sensitivity is desired.

Finally, since Peptizer is an open source project, and since Agents, Aggregators and profile configurations can be easily shared and implemented, we hope to establish an active user community at our purpose-built community portal (see <http://sites.google.com/site/peptizer>) that will continue to enhance the reach and power of the tool by adding Agents and progressively refined Aggregators, as well as by expanding its applicable scope to the output of many other search engines available today.

Acknowledgements

K.H. is supported by a Ph.D grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). L.M. would like to thank Rolf Apweiler and Henning Hermjakob for their support, and gratefully acknowledges the "ProDaC" grant LSHG-CT-2006-036814 of the European Union for funding his work. The lab in Ghent acknowledges support of research grants from the Fund for Scientific Research – Flanders (Belgium) (project numbers G.0156.05, G.0077.06 and G.0042.07), the Concerted Research Actions (project BOF07/GOA/012) from the Ghent University, the Interuniversity Attraction Poles (IAP-Phase VI, research project P6/28) and the European Union Interaction Proteome (6th Framework Program).

References

1. Domon, B., and Aebersold, R. (2006) Mass spectrometry and protein analysis. *Science* 312, 212-217.
2. Martens, L., and Hermjakob, H. (2007) Proteomics data validation: why all must provide data. *Molecular bioSystems* 3, 518-522.
3. Matthiesen, R. (2007) Methods, algorithms and tools in computational proteomics: A practical point of view. *Proteomics* 7, 2815-2832.
4. Eng, J. K., McCormack, A. L., and Yates, J. R. (1994) 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* 5, 976-989.
5. Perkins, D. N., Pappin, D. J., Creasy, D. M., and Cottrell, J. S. (1999) Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* 20, 3551-3567.
6. Craig, R., Cortens, J. P., and Beavis, R. C. (2004) Open source system for analyzing, validating, and storing protein identification data. *J Proteome Res* 3, 1234-1242.
7. Matthiesen, R., Trelle, M. B., Hojrup, P., Bunkenborg, J., and Jensen, O. N. (2005) VEMS 3.0: algorithms and computational tools for tandem mass spectrometry based identification of post-translational modifications in proteins. *Journal of proteome research* 4, 2338-2347.
8. Geer, L. Y., Markey, S. P., Kowalak, J. A., Wagner, L., Xu, M., Maynard, D. M., Yang, X., Shi, W., and Bryant, S. H. (2004) Open mass spectrometry search algorithm. *Journal of proteome research* 3, 958-964.
9. Sadygov, R. G., Cociorva, D., and Yates, J. R., 3rd (2004) Large-scale database searching using tandem mass spectra: looking up the answer in the back of the book. *Nature methods* 1, 195-202.
10. Brosch, M., Swamy, S., Hubbard, T., and Choudhary, J. (2008) Comparison of mascot and X!tandem performance for low and high accuracy mass spectrometry and the development of an adjusted mascot threshold. *Mol Cell Proteomics*.
11. Li, F., Sun, W., Gao, Y., and Wang, J. (2004) RScore: a peptide randomness score for evaluating tandem mass spectra. *Rapid Commun Mass Spectrom* 18, 1655-1659.
12. Savitski, M. M., Nielsen, M. L., and Zubarev, R. A. (2005) New data base-independent, sequence tag-based scoring of peptide MS/MS data validates Mowse scores, recovers below threshold data, singles out modified peptides, and assesses the quality of MS/MS techniques. *Mol Cell Proteomics* 4, 1180-1188.
13. Keller, A., Nesvizhskii, A. I., Kolker, E., and Aebersold, R. (2002) Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Analytical chemistry*, pp. 5383-5392.
14. Zhan, X., and Desiderio, D. M. (2006) Nitroproteins from a human pituitary adenoma tissue discovered with a nitrotyrosine affinity column and tandem mass spectrometry. *Analytical biochemistry* 354, 279-289.
15. Zhan, X., Du, Y., Crabb, J. S., Gu, X., Kern, T. S., and Crabb, J. W. (2007) Targets of tyrosine nitration in diabetic rat retina. *Mol Cell Proteomics*.
16. Hardwidge, P. R., Rodriguez-Escudero, I., Goode, D., Donohoe, S., Eng, J., Goodlett, D. R., Aebersold, R., and Finlay, B. B. (2004) Proteomic analysis of the intestinal epithelial cell response to enteropathogenic *Escherichia coli*. *The Journal of biological chemistry* 279, 20127-20136.

17. Veenstra, T. D., Conrads, T. P., and Issaq, H. J. (2004) What to do with "one-hit wonders"? *Electrophoresis* 25, 1278-1279.
18. Eddes, J. S., Kapp, E. A., Frecklington, D. F., Connolly, L. M., Layton, M. J., Moritz, R. L., and Simpson, R. J. (2002) CHOMPER: a bioinformatic tool for rapid validation of tandem mass spectrometry search results associated with high-throughput proteomic strategies. *Proteomics* 2, 1097-1103.
19. Tabb, D. L., McDonald, W. H., and Yates, J. R., 3rd (2002) DTASelect and Contrast: tools for assembling and comparing protein identifications from shotgun proteomics. *Journal of proteome research* 1, 21-26.
20. Pouillet, P., Carpentier, S., and Barillot, E. (2007) myProMS, a web server for management and validation of mass spectrometry-based proteomic data. *Proteomics* 7, 2553-2556.
21. Choi, H., and Nesvizhskii, A. I. (2008) Semisupervised model-based validation of Peptide identifications in mass spectrometry-based proteomics. *Journal of proteome research* 7, 254-265.
22. Gevaert, K., Van Damme, P., Ghesquiere, B., Impens, F., Martens, L., Helsens, K., and Vandekerckhove, J. (2007) A la carte proteomics with an emphasis on gel-free techniques. *Proteomics* 7, 2698-2718.
23. Stahl-Zeng, J., Lange, V., Ossola, R., Eckhardt, K., Krek, W., Aebersold, R., and Domon, B. (2007) High sensitivity detection of plasma proteins by multiple reaction monitoring of N-glycosites. *Mol Cell Proteomics* 6, 1809-1817.
24. Staes, A., Van Damme, P., Helsens, K., Demol, H., Vandekerckhove, J., and Gevaert, K. (2008) Improved recovery of proteome-informative, protein N-terminal peptides by combined fractional diagonal chromatography (COFRADIC). *Proteomics* 8, 1362-1370.
25. Gevaert, K., Van Damme, P., Martens, L., and Vandekerckhove, J. (2005) Diagonal reverse-phase chromatography applications in peptide-centric proteomics: ahead of catalogue-omics? *Analytical biochemistry* 345, 18-29.
26. Staes, A., Timmerman, E., Van Damme, J., Helsens, K., Vandekerckhove, J., Vollmer, M., and Gevaert, K. (2007) Assessing a novel microfluidic interface for shotgun proteome analyses. *Journal of separation science* 30, 1468-1476.
27. Martens, L., Vandekerckhove, J., and Gevaert, K. (2005) DBToolkit: processing protein databases for peptide-centric proteomics. *Bioinformatics (Oxford, England)* 21, 3584-3585.
28. Martens, L., Hermjakob, H., Jones, P., Adamski, M., Taylor, C., States, D., Gevaert, K., Vandekerckhove, J., and Apweiler, R. (2005) PRIDE: the proteomics identifications database. *Proteomics* 5, 3537-3545.
29. Elias, J. E., and Gygi, S. P. (2007) Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature methods* 4, 207-214.
30. Helsens, K., Martens, L., Vandekerckhove, J., and Gevaert, K. (2007) MascotDatfile: an open-source library to fully parse and analyse MASCOT MS/MS search results. *Proteomics* 7, 364-366.
31. Piggee, C. (2008) LIMS and the art of MS proteomics. *Analytical chemistry* 80, 4801-4806.
32. Ian, H. W., and Eibe, F. (2005) *Data Mining: Practical machine learning tools and techniques*, 2 Ed., Morgan Kaufmann, San Fransisco.
33. Nesvizhskii, A. I., and Aebersold, R. (2005) Interpretation of shotgun proteomic data: the protein inference problem. *Mol Cell Proteomics* 4, 1419-1440.

34. Kapp, E. A., Schutz, F., Reid, G. E., Eddes, J. S., Moritz, R. L., O'Hair, R. A., Speed, T. P., and Simpson, R. J. (2003) Mining a tandem mass spectrometry database to determine the trends and global factors influencing peptide fragmentation. *Analytical chemistry* 75, 6251-6264.
35. Tabb, D. L., Friedman, D. B., and Ham, A. J. (2006) Verification of automated peptide identifications from proteomic tandem mass spectra. *Nature protocols* 1, 2213-2222.

Figure and Table Legends

Figure 1: The voting mechanism of peptizer.

Peptide identifications are judged by a voting panel that consists of a series of Agents. Each Agent individually inspects a peptide identification and casts a vote that reflects whether or not the Agent requirements were fulfilled. The votes are all aggregated in a final score which is used to classify the peptide identification as either good or suspicious. The latter category is significantly enriched for false positives. Aggregation here is performed by simple summation of individual Agent scores.

Figure 2: Peptizer configuration using the graphical user interface.

The top panel is used to select the source for the identifications, while the central table serves to configure the Agents. Each Agent can be selected for inclusion in the voting panel, given veto rights, and, if applicable, its parameters can be set. Hovering over an Agent will pop up a tooltip explaining its workings. The panel below this table allows the selection of the aggregation method. The buttons to the right side of the central table can be used to save the current Agent configuration, or to load an existing one. The buttons at the bottom allow the complete configuration to be saved or loaded, and contains the button to start the task.

Figure 3: The Peptizer manual validation environment.

The tree structure on the left serves to navigate through the selected peptide identifications. Each tree node holds a spectrum with its confident peptide identifications. Unfolding the tree node shows the peptide sequences that were confidently assigned. By double-clicking a node, a new tab is opened on the right that shows a detailed view composed of an annotated sequence, an interactive spectrum viewer, and a data table. Each row in this data table shows a distinct

type of general or Agent-derived information, while each column represents a distinct confident peptide that was identified from the spectrum. The observed fragment ions for the selected peptide are annotated on the spectrum viewer, and on the annotated modified sequence.

Table 1: Agent configuration.

The listed agents and their parameters represent the voting panel that was configured to select suspicious peptide identifications from the example dataset.

Table 2: Summary of experimental peptizer usage.

(A) By applying peptizer the peptide identifications are separated in a good and suspicious set. This suspicious set is either completely discarded in full automatic validation mode or is further examined by the user in the semi automatic, manual validation mode. In the latter, identifications from the suspicious set will either be accepted or rejected by the user. The results of the manual validation are shown in (B).

Tables

Table 1

Agent	Veto	Parameter	Vote
Deamidation	TRUE	count : 2	Declines if 2 or more deamidations ¹
Suspect Residue	TRUE	sites : R;H	Declines if a His or internal Arg residue is present ²
Delta threshold	FALSE	delta : 10	Declines if score delta between ionscore and identity threshold is more then 10
Free NH2	FALSE	NA	Declines if N-terminus is unmodified
Homology	FALSE	NA	Declines if ion score or identity threshold is beyond the homology threshold
Length	FALSE	length : 9	Declines if the peptides has less then 9 amino acids
More Confident Hits	FALSE	delta : 20	Decline if there is more then one confident identification
N Term Acetylation	FALSE	NA	Recommends if the N-terminus is acetylated ³
Proline Peak	FALSE	intensity : 0.4	Decline absence of intense fragment ion N-terminal to an internal proline residue
b-ion coverage	FALSE	percentage : 0.10	Declines if b-ion coverage is less then 10%
y-ion coverage	FALSE	percentage : 0.25	Declines if y-ion coverage is less then 25%
Start Site	FALSE	low : 2 high : 200	Recommends if the peptide starts at protein position 1 or 2, declines if above protein position 200 and reserves in between

¹ When using MS/MS spectra obtained with low resolution mass spectrometers we typically enable deamidation as a variable modification to recover peptide identifications when the second isotope (not the monoisotopic ion) was selected for fragmentation. This modification tends to occur more frequently in false positive peptide identifications creating isobaric amino acid combinations a.o.

² Peptides that contain an internal basic residue were here suspicious since they should have been retained on the SCX column during sample preparation (**24**).

³ The N-terminal COFRADIC procedure includes an amino-acetylation step prior to digestion and about 95% of all identified peptides isolated by this procedure are alpha-N-acetylated. Such acetylated peptides are less likely to be false positives since they are simply more likely to occur. For the same reason, peptides that start in protein position 1 or 2 (methionine removal) are more likely to occur in the "true dataset".

Table 2

A	False positives in complete data set	False positives in accepted subset	False positives in discarded subset	Percentage of false positives removed	Percentage of true positives removed
Full automatic validation	4.0%	0.7%	48.7%	83.9%	3.7%
Semi automatic manual validation	4.0%	0.8%	57.0%	80.4%	2.5%
B	Identifications to validate, as selected by Peptizer	False positives accepted by user	False positives rejected by user	True positives accepted by user	True positives rejected by user
Estimated manual validation results	193	4	90	31	68

Figures

Figure 1

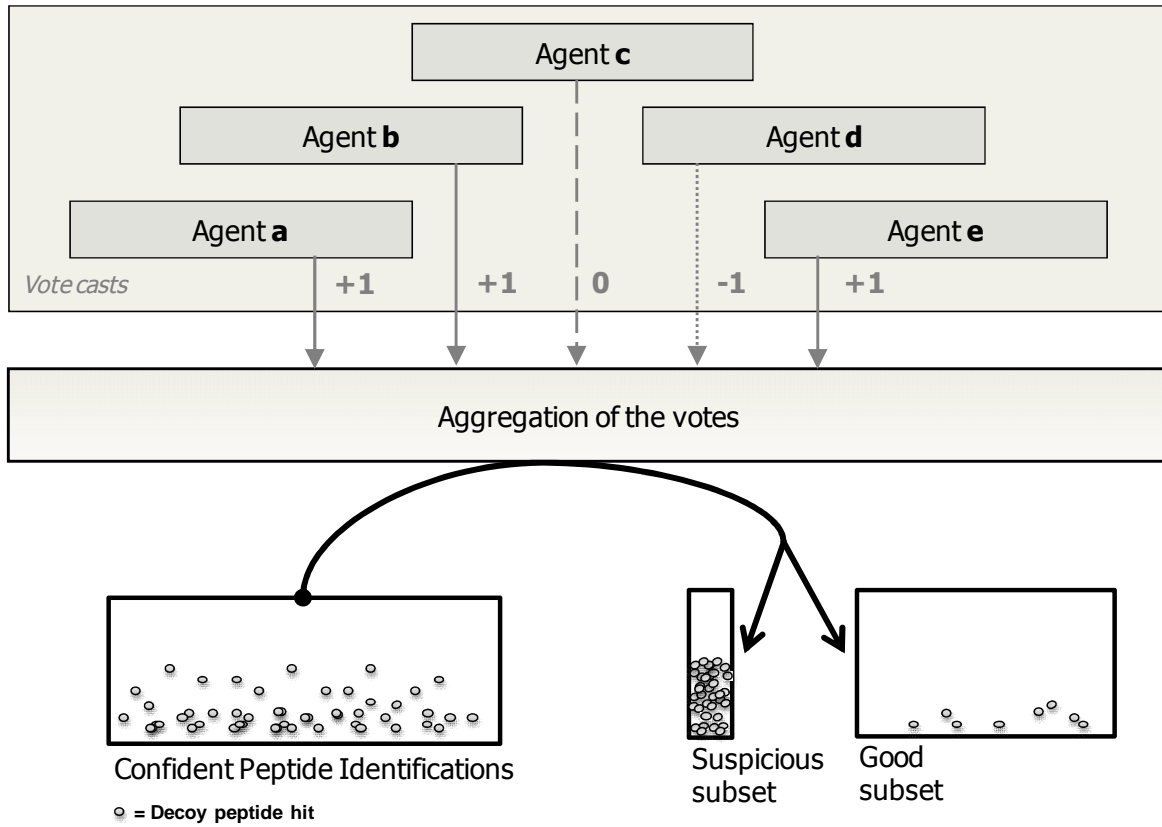


Figure 2

1. Data Source

Mascot dat File Memory Index C:\Temp\F001343.dat

2. Agent Summary Table

Name	Active	Veto	Parameters
Deamidation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	count : 2
Delta threshold	<input checked="" type="checkbox"/>	<input type="checkbox"/>	delta : 10
Free NH2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NA
Homology	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NA
Length	<input checked="" type="checkbox"/>	<input type="checkbox"/>	length : 9
More Confident Hits	<input checked="" type="checkbox"/>	<input type="checkbox"/>	delta : 20
N Term Acetylation	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NA
Proline Peak	<input checked="" type="checkbox"/>	<input type="checkbox"/>	intensity : 0.4
Sequence RegExp	<input type="checkbox"/>	<input type="checkbox"/>	regular expression : <code>*[il]3.*</code>
Start Site	<input checked="" type="checkbox"/>	<input type="checkbox"/>	low : 2 high : 200
Suspect Residue	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	sites : R;H
b-ion coverage	<input checked="" type="checkbox"/>	<input type="checkbox"/>	percentage : 0.10
y-ion coverage	<input checked="" type="checkbox"/>	<input type="checkbox"/>	percentage : 0.25

Inspects for the length of the peptide. Scores +1 if the peptide is smaller then the given length (9). Scores 0 if more.

3. AgentAggregator Selection Table

Best Hit Agent Aggregator **Properties Table**

Name	Value
Theshold Score	2

Confidence (input alpha value 0.05, results in 95% confidence)

Figure 3

