RESEARCH ARTICLE

ProteoAnnotator – Open source proteogenomics annotation software supporting PSI standards

Fawaz Ghali[1], Ritesh Krishna[1]*, Simon Perkins[1], Andrew Collins[1], Dong Xia[2], Jonathan Wastling[2,3] and Andrew R. Jones[1]*

[1] Institute of Integrative Biology, University of Liverpool, Liverpool, UK
[2] Department of Infection Biology, Institute of Infection and Global Health, University of Liverpool, Liverpool, UK
[3] Health Protection Research Unit in Emerging and Zoonotic Infections, The National Institute for Health Research, University of Liverpool, Liverpool, UK

Received: June 10, 2014
Revised: September 10, 2014
Accepted: October 2, 2014

The recent massive increase in capability for sequencing genomes is producing enormous advances in our understanding of biological systems. However, there is a bottleneck in genome annotation – determining the structure of all transcribed genes. Experimental data from MS studies can play a major role in confirming and correcting gene structure – proteogenomics. However, there are some technical and practical challenges to overcome, since proteogenomics requires pipelines comprising a complex set of interconnected modules as well as bespoke routines, for example in protein inference and statistics. We are introducing a complete, open source pipeline for proteogenomics, called ProteoAnnotator, which incorporates a graphical user interface and implements the Proteomics Standards Initiative mzIdentML standard for each analysis stage. All steps are included as standalone modules with the mzIdentML library, allowing other groups to re-use the whole pipeline or constituent parts within other tools. We have developed new modules for pre-processing and combining multiple search databases, for performing peptide-level statistics on mzIdentML files, for scoring grouped protein identifications matched to a given genomic locus to validate that updates to the official gene models are statistically sound and for mapping end results back onto the genome. ProteoAnnotator is available from http://www.proteoannotator.org/. All MS data have been deposited in the ProteomeXchange with identifiers PXD001042 and PXD001390 (http://proteomecentral.proteomexchange.org/dataset/PXD001042; http://proteomecentral.proteomexchange.org/dataset/PXD001390).

Keywords:
mzIdentML / Open source / ProteoAnnotator / Proteogenomics / Proteomics Standards Initiative

Additional supporting information may be found in the online version of this article at the publisher's web-site

1 Introduction

Accurate gene annotation is a challenging stage in genome sequencing pipelines. The rapidly lowering cost of next-generation sequencing makes it difficult for genome annotation pipelines to keep pace. Gene annotation can be achieved either manually or via automated pipelines. While manual annotation of protein-coding genes is usually more reliable, it may not always be feasible due to time constraints. Therefore, genome annotations are mostly based on predictions, where traditional gene annotations software pipelines use external evidence to enhance the accuracy of the gene models of any organism (reviewed in [1]). Gene-finding software often

Correspondence: Dr. Andrew Robert Jones, Institute of Integrative Biology, Biosciences Building, University of Liverpool, Liverpool L69 7ZB, UK
E-mail: andrew.jones@liv.ac.uk

Abbreviations: CSV, comma-separated values; **FDR**, false discovery rate; **MGF**, MASCOT generic format; **PSI**, Proteomics Standards Initiative; **PSM**, peptide spectrum match

*These authors contributed equally to this work.
Colour Online: See the article online to view Figs. 1–3 in colour.

introduces errors in annotating the gene models of Eukaryotic organisms, since the correct prediction of splice sites and the position of the first exon do not have motifs that can be easily predicted. Various experimental techniques are used to provide supporting evidence for exon–intron structure, such as ESTs and mRNA transcript sequencing (RNA-Seq), including evidence for alternative splicing of transcripts [2]. Most genome projects now incorporate some transcript sequencing to facilitate annotation. However, all RNA-based methods for annotation have the drawback that they cannot provide direct evidence that a putative splice product is translated into a genuine protein molecule in a given biological system.

In the context of proteomic investigations, LC-MS can produce protein identifications and quantitative data on a large scale. In a classic 'shotgun' pipeline, MS data are collected from peptides resulting from proteolysis of the total protein pool. Most commonly, modern instruments function in two steps (MS/MS) where peptides are fragmented in the second step (MS^2) and each MS^2 spectrum is queried by a search engine (such as MASCOT [3], OMSSA [4], X!Tandem [5] or MSGF+ [6]) against a protein sequence database to make a peptide spectrum match (PSM). MS data can be used to annotate the genome of any organism, in theory providing evidence for an isoform resulting from predicted splicing, for the correct start codon of a gene, as well as single amino acid polymorphisms and PTMs. However, proteogenomics is a challenging task – as genome annotations ('official' gene model sets) change on a regular basis – often once per year, rendering proteomics identifications to previous releases out-of-date and requiring a re-analysis with respect to the new models. There are also technical and practical challenges, including the requirement for a complex interconnected pipeline of modules often not designed with proteogenomics in mind, and often proving difficult to integrate due to file format or design incompatibility.

Various groups have produced software for proteogenomics. An early initiative in this direction was the genome annotating proteomic pipeline [7], using the open source X!Tandem software to query against particular genome builds from a relational database. A recent tool Peppy [8] performs the most common proteogenomics tasks, such as generating a peptide database from a genome, tracking peptide loci, matching peptides to MS/MS spectra and performing false discovery rate (FDR) analysis. The PG Nexus [9] allows users to visualise peptides in the context of genomes – done in the Integrated Genome Viewer. PG Nexus is integrated into the Galaxy cloud environment [10] and is available in the Galaxy tool shed. Genomics and transcriptomics data sets can be used as custom sequence database in MASCOT searches. The Samifier tool [9] then converts the output results from MS/MS searches into a .SAM file format that can be visualised in the Integrative Genomics Viewer. The iPiG tool [11] integrates peptide identifications from MS data into existing genome browser visualisations. It also supports input (but not output) of the mzIdentML standard for the identified peptides. However, the tool does not perform post-processing, it relies on prior FDR estimation methods.

While database search algorithms are most commonly used in this space, they are limited to identifying a priori predicted sequences present in the protein database. Another approach is de novo sequencing that does not require a protein database, but there is a general consensus that purely de novo approaches exhibit overall weaker performance (lower sensitivity vs. specificity) than those based on sequence database search. However, tools such as GenoMS [12] use the combined strengths of database and de novo methods. In GenoMS, the database search tool InsPecT [13] is used to identify protein sequence templates. Then, these templates are used to sequence de novo regions of the target protein that are missing or diverged from the database.

Recent work by Castellana et al. [14] queried MS data against a combined six frame translation and a *splice graph* (formed from experimental mRNA sequences mapped to the genome as well as predicted splice sites), followed by genomic clustering of PSMs to identify regions likely to contain genes missed in the annotation. The study was able to identify a large number of novel genes and updates to existing genes.

To our knowledge, there is no single pipeline that is automated (requiring almost no complex setup procedures or parameterisation), incorporates bespoke algorithms for genomic loci identification and scoring and performs the common tasks for a proteogenomics pipeline using Proteomics Standards Initiative (PSI) for each analysis step. Therefore, we are introducing an open source pipeline for proteogenomics, called ProteoAnnotator. ProteoAnnotator has a simple setup procedure, a graphical interface for end users (such as laboratory scientists) or command line mode for informatics groups wishing to run it in a parallel environment. ProteoAnnotator is implemented using the PSI mzIdentML standard data format [15] for peptide and protein identifications, and an associated library of routines [16]. By using mzIdentML, individual modules of ProteoAnnotator can be incorporated into other tools and the outputs can also be directly submitted to the ProteomeXchange central database [17] and PRIDE [18]. ProteoAnnotator can be downloaded as a single, self-contained zip archive from http://www.proteoannotator.org/.

2 Materials and methods

2.1 ProteoAnnotator structure and modules

ProteoAnnotator comprises a set of modules that can be used as individual tools or as a combined pipeline for genome annotation. It can be run in two different modes: command line mode and graphical user interface mode. ProteoAnnotator is designed to be user-friendly from the installation phase to running with a minimum effort. Moreover, ProteoAnnotator is designed using a modular structure that allows bioinformatics groups to easily use, run and adapt the pipeline to match their need. ProteoAnnotator is released under the

Apache 2.0 licence so that other groups can use the code, modules or the complete pipeline in other projects without restriction.

The ProteoAnnotator pipeline (Fig. 1) takes GFF3 (genome coordinates) and FASTA (protein sequence) file formats as search database inputs. The FASTA file is optional if the GFF file already contains the FASTA-formatted data within the file (since this is optional in GFF3 format). The design of ProteoAnnotator requires that the user uploads one set of genomic coordinates and protein sequences as the current 'official' models for the genome of interest – these are flagged as the 'A' gene/protein set for further analysis steps. In case the genome has recently been sequenced with no official models, the user should upload the set of gene models considered to be the most high quality, for example as predicted by gene-finding software. The user then has the option to upload an unlimited number of additional sets of predicted gene models in order of quality preference – flagged as 'B', 'C', 'D' set and so on. The pipeline then produces a single concatenated FASTA file. ProteoAnnotator uses the MzidLib routines [16] for pre- and post-processing and SearchGUI [19] for creating a decoy databases and running the MS/MS search using Omssa and X!Tandem (SearchGUI is an open source tool that allows running a search using different open source search engines). Figure 1 shows the ProteoAnnotator workflow and modules used. While some modules are already being published and used [16], a new set of modules were developed and added to the MzidLib specifically for ProteoAnnotator and SearchGUI integration. Table 1 lists ProteoAnnotator command line parameters. A detailed description of each step of the ProteoAnnotator pipeline is as follows.

Step (a) *Creating generic FASTA files*: The FASTA format used as input to database search engines does not have a standardised header, therefore this stage gives a uniformly structured protein accession and description to ensure the search engines embedded in SearchGUI function correctly.

Step (b) *Search database concatenation*: ProteoAnnotator incorporates multiple search databases generated by gene finding software or derived by assembly from RNASeq data, to be compared versus the official gene set. This step concatenates multiple databases, adding a prefix to the accessions from each input set in order of database preference (A, B, C, D, and so on), which is picked up by the protein-grouping algorithm downstream.

Step (c) *Creating a decoy database*: This step is performed by SearchGUI [19] to create a concatenated target-decoy database, containing reversed target sequences, used for FDR calculation.

Step (d) *Running the MS/MS search*: The search is done via SearchGUI. In ProteoAnnotator we use OMSSA and X!Tandem, although SearchGUI also incorporates MSGF+ and MS-Amanda (http://ms.imp.ac.at/?goto=msamanda) that may be incorporated into later builds of ProteoAnnotator if testing demonstrates superior performance from their inclusion.

Step (e) *Format conversion of search outputs*: The native file formats of OMSSA (OMX) and X!Tandem ("extensible markup language") are converted to mzidentML, using a module from MzidLib previously described [16].

Step (f) *Combine mzIdentML files*: If multiple MASCOT generic format (MGF) files are provided as input as a single analysis unit, the results can be combined prior to protein grouping. This step removes redundant protein and peptide entries, concatenating the PSMs within the file – keeping references to the source MGF file intact internally. Each individual file is limited to <1 GB size and <25 000 scans since SearchGUI automatically splits files above this limit, and it becomes harder to trace data through the ProteoAnnotator. In a future build, we will add automated splitting of files if they are above the limit.

Step (g) *Combining search engines outputs*: This module rescores and combines PSMs from two or three search engine (OMSSA and X!Tandem only in the current build) to produce a single output, using the algorithm previously described [20].

Step (h) *Determining the FDR at the peptide-level*: Recent debate in the literature suggests that performing FDR analysis and thresholding at (say) 1% FDR on PSM can lead to a higher level of FDR for peptides (discussed in [21]). This is because highly abundant (true positive) peptides are often observed in multiple spectra – due to the peptide eluting off a column for a period of time and being fragmented on many occasions. False-positive peptide identifications are often random events observed only once. As an example, in a set of 1000 peptide identifications with 1% PSM FDR (1000 target PSMs, ten decoy PSMs), one might observe 300 different target peptides and ten decoy peptides – leading to *peptide-level* FDR of 3.3%. We have added a new routine to the MzidLib for performing peptide-level statistics, taking the best scoring PSM per peptide (ignoring different modification and charge states), and re-applying the algorithms described in [20]. This typically produces more conservative estimates of FDR (and fewer peptides) to be passed on to the protein-grouping stage. Note, there are different methods of grouping PSMs per peptide, for example taking into account modification states and/or charge states of a peptide as different entities, which may be implemented in ProteoAnnotator in due course if we find demand for these methods.

Step (i) *Peptide-level thresholding*: The MzidLib [16] contains a routine that allows an attribute to be set for each PSM in the mzIdentML file *passThreshold = true or false*, based on a given PSM-level score. We have adapted the routine to account for peptide-level thresholding, setting passThreshold = true for only the best scoring PSM per peptide that passes the threshold (1% peptide-level FDRScore is recommended for use in ProteoAnnotator, but users can alter this value).

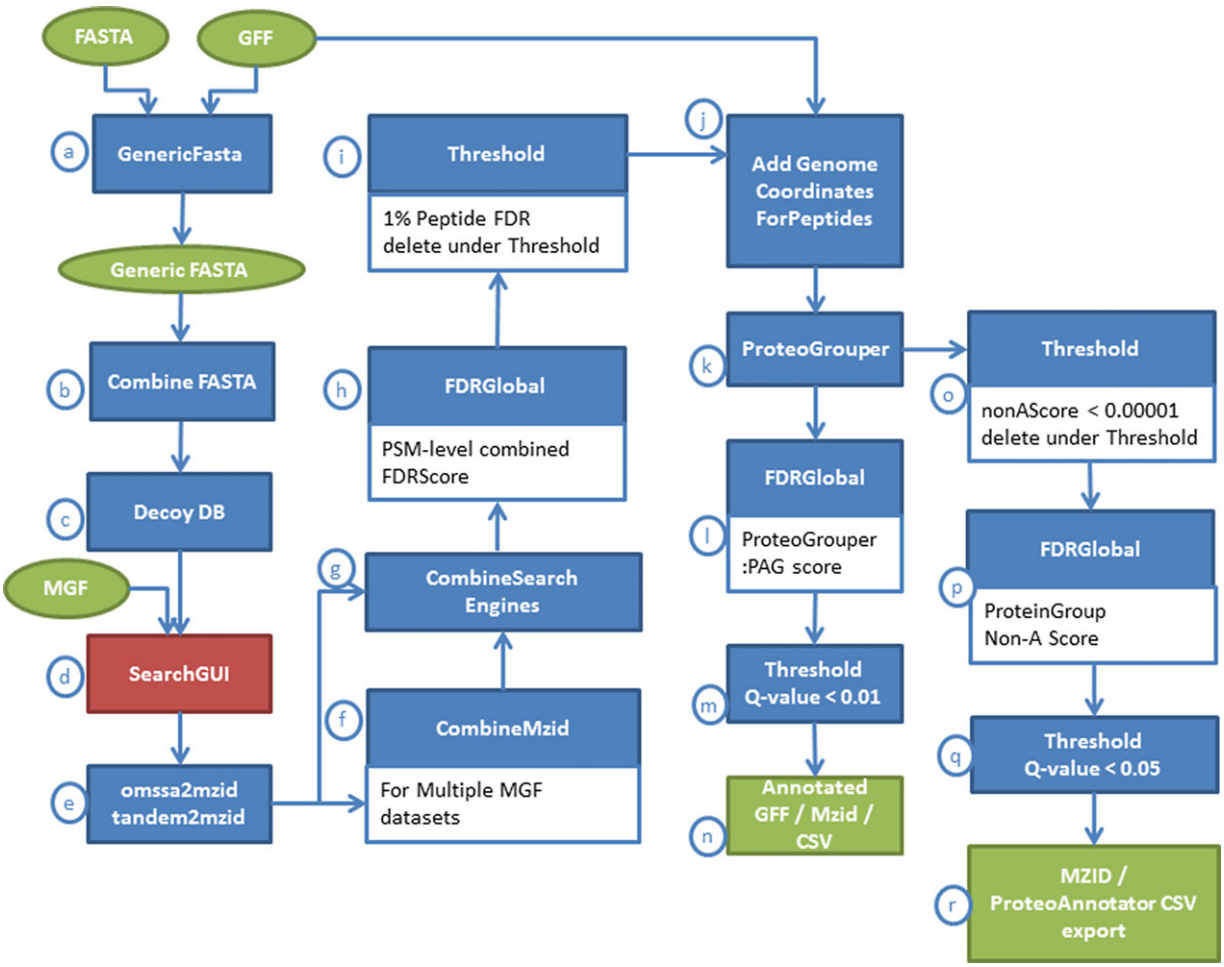

Figure 1. The ProteoAnnotator workflow, as implemented in the MzidLib (blue rectangles), with file inputs (green ovals), outputs (green rectangles) and SearchGUI integration (red rectangle). Steps (a–r) are explained in the Section 2.

Step (j) *Adding genome coordinates to the mzIdentML file*: This module maps the positions (start/end within the protein) of putatively identified peptides back onto the genome coordinate system (including mapping peptides across multiple exons, as necessary), as specified in the GFF file.

Step (k) *Protein inference*: The ProteoGrouper algorithm is described in [16], but has been adapted with a new scoring scheme for ProteoAnnotator. ProteoGrouper takes as input the set of peptides confidently identified, based on the threshold applied in Step (i). Peptide-level, protein-level and protein-group level scoring are calculated as follows:

Each peptide contributing to protein identification is given a peptide score (*pepscore*) as follows:

$$\text{pepscore} = -10 - (10 \times \log_{10}(\text{peptide-level FDRScore})).$$

The *peptide level FDRScore* is an approximation of the local FDR associated with a given peptide-level identification (adapted from [20]) in ProteoAnnotator carrying values from >0 to <0.01 following thresholding at 1% FDR, for example. The equation converts values such that a peptide with *peptide-level FDR score* = 0.01 gives *pepscore* = 10, *peptide-level FDR score* = 0.001 gives *pep score* = 20 and so on, giving rise to human readable score for protein-level identifications in the following step.

Each protein (accession) identified is given a score as follows 'ProteoGrouper:PDH score' (with accession MS:1002235 in the PSI-MS controlled vocabulary [CV] [22]):

$$\text{PDH score} = \sum_{i=1}^{n} \text{pepscore}_i,$$

where n is the total number of peptides mapped to a given protein accession, i is the label given to each peptide and pepscore_i is the pepscore for peptide i. The set of pepscores includes all the peptides that can be mapped to a given protein, regardless of the number of other proteins to which they can also be mapped.

ProteoGrouper assigns all protein identifications to groups, based on set relationships of peptides (accounting

Table 1. The ProteoAnnotator command line parameters

Parameter	Optional/mandatory	Explanation
-prefix	Optional	A prefix to be attached to the output file names
-inputGFF_A	Mandatory	The canonical GFF file
-inputFasta_A	Optional if the canonical GFF contains the FASTA	The protein database
-outputFolder	Mandatory	The output folder for the analysis
-spectrum_files	Mandatory	The MGF files to be searched
-searchParameters	Mandatory	The search parameters file to be used for the search in a text file, following the format required for SearchGUI
-inputPredicted	Optional	The non-canonical gene models, these are a set of GFF/FASTA files. The GFF and FASTA are separated by ';' and the pairs are separated by '##'
-peptideThreshValue	Mandatory	The threshold implemented for peptide-level FDR (0.01 is the recommended value and default in the interface)
-proteinThreshValue	Mandatory	The threshold implemented for protein group level FDR (0.01 is the recommended value and default in the interface)

for isoleucine/leucine ambiguity) as described in [16]. Each protein group is assigned a score 'ProteoGrouper:PAG score' (PSI-MS CV identifier is MS:1002236) as follows:

$$\text{PAG score} = \sum_{r=1}^{n_r} \text{pepscore}_r + \sum_{u=1}^{n_u} \text{pepscore}_u,$$

where n_r and n_u are the total number of razor and unique peptides, respectively, mapped to the group leader of the protein group.

The PAG score is based on summing the scores for peptides classified as 'unique' within the group (can only be mapped to a lead protein within the group) or 'razor' peptides (the lead protein within the group has been assigned the razor peptides as having more evidence for its identification than any other protein in the list). The 'ProteoGrouper:PAG score' can be used to determine the strength of evidence for a given protein group to have been identified, relative to other groups in the overall list. In ProteoAnnotator, a new score is calculated by the module 'ProteoAnnotator:non-canonical gene model score' (term identifier MS:1002474), which has been added to the PSI-MS controlled vocabulary [22], so that it is valid for use within mzIdentML. The non-canonical gene model score is calculated as follows:

$$\text{noncanonical gene model score} = \sum_{A=1}^{n_A} \text{pepscore}_A,$$

where n_A is the total number peptides that do not map to 'A' (official) gene models within the protein group.

The score is calculated by summing the scores for individual peptides that have been identified in a given protein group, but which have not been identified in an official (A) gene model (in any protein group), and thus provide evidence that the annotation can be improved for a given predicted locus. All decoy peptide identifications (mapped into decoy protein groups) are included in this score calculation (since by definition they are not mapped to the official gene models). When the protein group list is ordered by the non-canonical gene model score (Step p), the decoy protein groups give a conservative background distribution against which any loci determined to have evidence for improvements in the genome annotation to be compared.

The ProteoAnnotator pipeline splits down to two pathways ('l, m, n' and 'o, p, q, r') to provide users with two different types of output. The former provides evidence that given proteins and peptides originating from the official gene model and/or alternative models have been identified. The latter provides evidence that the official genome annotation has the potential to be improved for loci confidently identified.

Step (l) *FDR calculation for protein groups*: The MzidLib: FDR module is applied to order protein group identifications, using the *PAG score* to order the list of target and decoy identifications.

Step (m) *Threshold for protein groups*: A threshold is applied at the protein group level, setting passThreshold = 'true' on those protein groups with *q*-value less than 0.01 (e.g. 1% protein group level FDR is recommended for use in ProteoAnnotator, but users can alter this value).

Step (n) *File format* exports: This step exports results to various file formats, including a fully annotated mzIdentML file (suitable for submission to a public repository such as PRIDE/ProteomeXchange), a GFF3 file annotated with peptide coordinates for visualisation in a genome browser and various comma-separated values (CSV) file formats. The following CSV exported files are produced: (i) a file containing one row of data (including the details for the representative protein) per protein group – suffix 'exportRepProteinPerPAGOnly.csv', (ii) a file containing one row per PSM identified – suffix 'exportPSMs.csv', (iii) a file containing one row per protein accession that has been assigned to any group in the protein list – suffix 'exportProteinsOnly.csv', (iv) a file containing one row per PSM

supporting each protein accession in the protein list, along with protein-level scoring and protein group assignment (including redundancy in PSM to protein mapping) – suffix 'exportProteinGroups.csv'. The number of rows (protein groups) in file (i) with passThreshold = true gives an indication of the overall number of loci that have been identified – regardless of the number of search databases included in Step (b). Each protein group is assigned a group leader (by ProteoGrouper), based on the protein within the group having most evidence, followed by alphabetical order in the case of ties. This means that if an 'A' (official) gene model-derived protein is identified with the same evidence set (peptides) as a 'B' or 'C' non-official gene model, the 'A' protein will be assigned as the group leader. Following this strategy, any protein group in which an 'A' protein is not the group leader indicates that at the given locus peptides have been identified (so called *alternative peptides*) that do not match the official gene models, and thus are candidate regions for improving the genome annotation. However, there is no statistical basis for assuming that such identifications are significant at the genomic locus level – since a single border-line peptide identification (say FDRScore close to 0.01), that is not matched to the 'A' models, could convey group leader status to a 'B' or 'C' model. Such peptides can be randomly generated false-positives, especially given that non-'A' databases could be large (e.g. generated from six frame translations). As such, this view of the data only gives an indication that a given locus may have evidence for improvements to the given official gene model. The statistical significance of matches to non-official gene models is handled by workflow path (Steps o–r) as follows.

Step (o) *Remove protein groups with no support for alternative gene models*: In this workflow path, the threshold module is applied to remove any protein groups that have a *non-canonical gene model score* equal to zero, that is the group contains no evidence supporting an alternative annotation at the given loci (all unique and razor peptides have been assigned to an 'A' gene model). This is implemented by deleting all protein groups from the result set that have *non-canonical gene model score* < 0.00001.

Step (p) *FDR analysis based on the non-canonical gene model score*: The FDR module is performed at the protein group level using the *non-canonical gene model score* for ordering all identifications (targets and decoys), calculating an estimate of global FDR and a *q*-value for each protein group.

Step (q) *Apply threshold based on non-canonical gene model score*: The threshold routine is applied to protein groups, setting passThreshold = true for protein groups with *q*-value <0.05. This threshold value cannot be altered currently in the graphical interface as we do not want to clutter the interface with too many parameters. No identifications are removed at this step, the only difference being which protein groups are flagged as passThreshold = 'true|false' in the downstream mzIdentML and CSV files. In the CSV view, users can easily perform different types of thresholding for their own purposes, simply by ordering results by non-canonical gene model score. The purpose of this step is to determine the statistical significance of loci determined to be carrying evidence for improvements to the genome annotation, which we call *alternative loci*. This step ensures a conservative calculation of *q* values, since the target loci themselves may be well supported (a high PAG score) but only the target peptides that do not match 'A' models contribute to the *non-canonical gene model score*, whereas all decoy peptides contribute to the score for decoy protein groups. The distribution thus gives a background of the rate of matches expected by chance to genomic loci in the large concatenated databases. We apply a *q*-value threshold of <0.05 rather than 0.01, since we know that the decoy distribution is conservative, and due to the granularity of the calculation. A typical data set may only have tens of loci carrying a *non-canonical gene model score* and thus the first decoy encountered in the ordered list would push the *q*-value estimate over the 0.01 threshold. Genomic loci with *q* values from 0.01 to 0.05 in this analysis should be treated with care, as the level of evidence for an update to the gene model annotation is clearly weak – usually based on a single peptide identifications without a strong score.

Step (r) *File format exports*: The results of this pathway are exported to mzIdentML and CSV files – which should be used by genome annotators to prioritise those loci where updates to the official genome annotation are most well supported by proteomic data. The CSV file (suffix 'exportProteoAnnotator.csv') contains one row per protein group identified, all of which carry some evidence for updates to the annotation at a particular loci. The 'A' gene model(s) present within each protein group are given in one cell (allowing annotators to most easily locate the genes with suggested improvements), along with the alternative peptides identified and the *non-canonical gene model score* for each group.

2.2 Validation and testing of ProteoAnnotator

The value of a proteogenomics pipeline for improving a genome annotation is determined by the availability of large MS data sets, mining deep into the proteome and the quality of existing 'official' gene models – as the quality improves, self-evidently the capability of improving the annotation decreases. Validating a proteogenomics pipeline is challenging, since it is difficult to find a data set in which there is a known 'ground truth' to test against, in this case 'ground truth' are the annotation improvements suggested by ProteoAnnotator – which are real proteins – missed or incorrectly annotated in the official gene set. As such, to validate the performance of the ProteoAnnotator, we have performed a historic analysis against two releases of official gene models for the Apicomplexan parasite *Toxoplasma gondii*, separated by several years in which manual curation and the incorporation of RNASeq transcriptome sequence data 'improved' the gene model set, in a process independent from our proteogenomics

Table 2. Case studies and data sets used to test the performance of ProteoAnnotator

Case study	Canonical gene models	Non-canonical gene models
1	TgondiiME49–6.0 (A model)	AUGUSTUS-6.0 (B model) Glimmer-6.0 (C model)
2	TgondiiME49–10 (A model)	AUGUSTUS-6.0 (B model) Glimmer-6.0 (C model)

analysis presented here. In this analysis, we downloaded the official gene sets from ToxoDB [23] for the *T. gondii* release 6 (2009) and release 10 (2014). We also downloaded the source genomic sequence *T. gondii* release 6 – which was used to predict alternate candidate identifications using two freely available gene-finding software packages – Augustus [24] and GlimmerHMM [25]. We used Augustus to predict a gene model set based on the *T. gondii* release 6 genome sequence using the following parameters – genome file: http://toxodb.org/common/downloads/release-6.0/Tgondii/TgondiiME49Genomic_ToxoDB-6.0.fasta; user set UTR prediction: false; report genes on: both strands; alternative transcripts: few; allowed gene structure: predict any number of (possibly partial) genes; ignore conflicts with other strand: false. Note that we experimented with several parameters, including 'Ignore conflicts with other strand: true' and 'Alternative transcripts: medium|many', which made little differences to the overall results (data not shown). For GlimmerHMM, we used the default parameters and the same input genomic DNA. We then performed two analyses, as shown in Table 2. The purpose of the analysis was to ask several questions. First, given that we know that *T. gondii* gene model annotations in release 6 were imperfect (many updates have since been made to the gene models) – does ProteoAnnotator predict a set of loci for which gene models can be improved (Case study 1)? Second, under the assumption that the release 10 gene set is now considered very high quality, the majority of the alternative peptides identified in Case study 1 should be mapped to official gene models in release 10 (Case study 2). If ProteoAnnotator is producing a large number of random and incorrect loci supported by alternative peptides, we should see a large number of non-canonical loci identified in both pipelines. Lastly, does ProteoAnnotator suggest that improvements can still be made to the release 10 data set (Case study 2)?

Toxoplasma gondii RH tachyzoites were separated by 1D SDS-PAGE on a 12% (v/v) acrylamide gel, from which 16 gel bands were excised and digested with trypsin. The digests were then pooled into eight samples for LC-MS/MS analysis. Peptide mixtures were analysed by online nano-flow LC using the nano-ACQUITY-nLC system (Waters MS Technologies, Manchester, UK) coupled to an LTQ-Orbitrap Velos (ThermoFisher Scientific, Bremen, Germany) mass spectrometer equipped with the manufacturer's nano-spray ion source.

The following parameters were set in ProteoAnnotator and passed to SearchGUI: precursor tolerance 5 ppm, fragment tolerance: 0.5 Da, fixed mods: carbamidomethylation on cysteine and variable modification of oxidation of methionine. Other parameters were left as defaults, as described at the SearchGUI website (https://code.google.com/p/searchgui/).

A further use of ProteoAnnotator (Case study 3) is to demonstrate mapping or automated re-analysis of a data set against a new or updated genome – including support for very large input data sets. To demonstrate this functionality, we have performed a re-analysis of a data set publicly accessible from ProteomeXchange central (identifier PXD000652) from a study on cerebrospinal fluid [26]. The original study looked at seven different sample groups, we have analysed a single sample group 'CSF, gel separated, depleted fraction' – consisting of 46 different LC-MS runs derived from 1D SDS-PAGE. The study used SearchGUI (as in ProteoAnnotator), but was searched against UniProt. We wished to demonstrate that ProteoAnnotator can be used to map to the relevant canonical genome sequence – in this case we searched against the current Ensembl Human build 76 [27], otherwise following the same search parameters as reported in [26].

2.3 ProteoAnnotator interface

ProteoAnnotator can be run within Proteosuite, which is a tool originally designed for quantitative proteomics, and now incorporating ProteoAnnotator (http://www.proteosuite.org/). The pipeline can be run in the GUI as follows: (i) raw MGF data are loaded into Proteosuite, and the genome annotation option is checked. Multiple MGF files may be selected, which will be combined following the search step and prior to CombineSearchEngines (Step (f); Fig. 1). (ii) Search parameters and ProteoAnnotator options are set via a form as shown in Fig. 2. Many of these options are directly passed on to SearchGUI. Users are expected to enter one official gene set, and optionally add several non-official gene sets, as GFF3 formatted files. GFF3 data for each gene model set are mandatory, however, a separate FASTA file containing protein sequences is only mandatory if the GFF3 file lacks protein sequences. Protein sequence data should contain only targets, no decoys – the pipeline takes care of decoy database generation. (iii) The pipeline is started using the 'Run' button, and the user is later notified when it is completed. The output of the pipeline may be inspected in the 'annotation_output' folder where the raw data were located. Output items of interest include newly annotated GFF3 file (for each gene model set entered), various mzIdentML files, various CSV files (as described above) and a log file called ProteoAnnotator.txt.

3 Results and discussion

The summary results for the two *T. gondii* case studies are presented in Table 3 and in Supporting Information Files

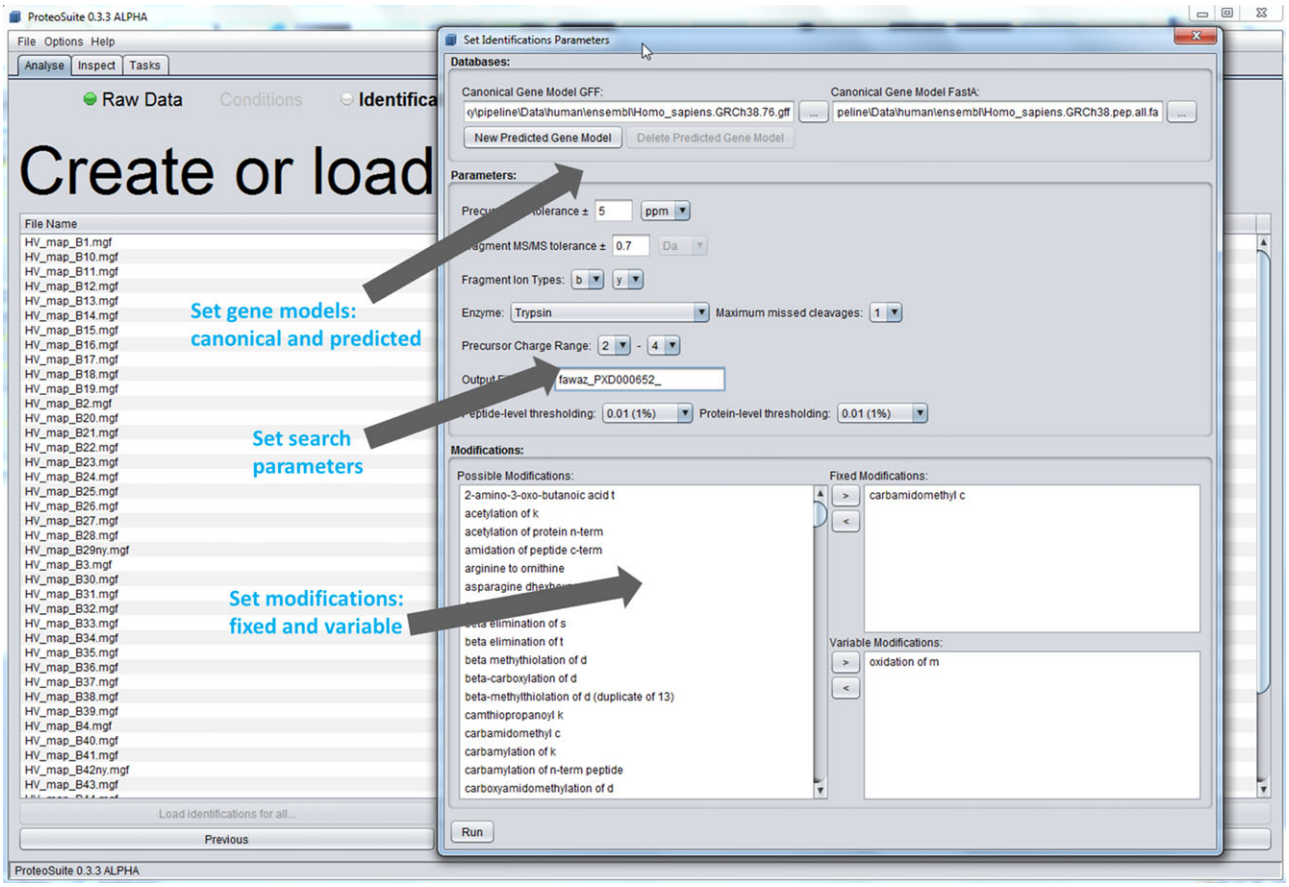

Figure 2. The ProteoAnnotator graphical user interface, as implemented in Proteosuite.

1–6. The results give confidence that ProteoAnnotator is functioning to produce coherent and statistically sound results. Overall, broadly the same number of protein groups and peptides are confidently identified in both case studies, indicating that the overall set and number of loci that can be identified has not radically altered between *T. gondii* release 6 (1619 protein groups) and release 10 (1611 protein groups) gene models. For comparison, when ProteoAnnotator is run solely with this data set against only the release 10 official models, 1571 protein groups are identified at $q < 0.01$ (data not shown). Indicating that even though the search database is several fold larger in the re-annotation case studies, there is no large increase in the overall number of proteins identified, which would suggest false-positive identifications are being made.

In comparing the results from Case study 1 and 2, we see a large difference in the number of alternative loci and alternative peptides. In Case study 1, we see 83 alternative loci passing the threshold, supported by 289 alternative peptides. In Case study 2, the number of alternative loci has dropped to 15, supported by only 35 alternative peptides. These results suggest that for the proteins identifiable from this data set (approximately 1600) in 2009, around 5% of the loci could be improved by the pipeline, falling to less than 1% in the 2014 release. We cannot determine that these 1600 gene models in release 10 are now all broadly correct, since these are

Table 3. Summary of results from the two case studies described in the methods

	Total protein groups identified at *q*-value (0.01)	Total peptides identified at *q*-value < 0.01	Total alternative loci identified at *q*-value < 0.05	Total alternative peptides within the alternative loci passing threshold
Case study 1	1619[S1]	10261[S2]	83[S3]	289[S3]
Case study 2	1611[S4]	10299[S5]	15[S6]	35[S6]

The superscripts relate to the Supporting Information Files (1–6) in which the evidence for each count is presented.

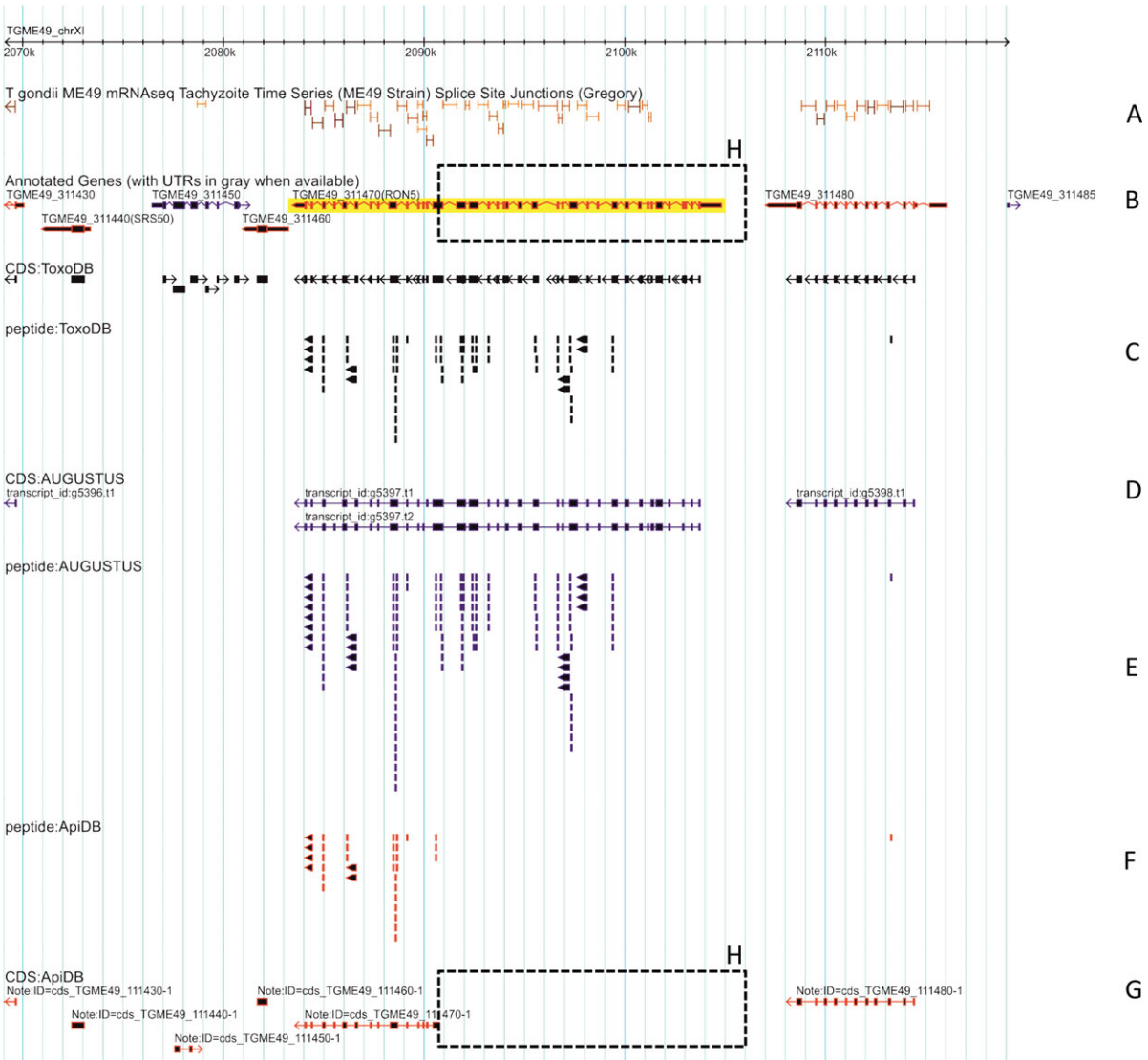

Figure 3. A visualisation of the top scoring alternative loci (Augustus prediction g5397.t1) in the ToxoDB genome browser. The data have been aligned with: (A) RNASeq-generated splice junctions mapped onto release 10 official genes; (B) the corresponding official gene in release 10; (C) the peptide mapped to this gene in Case study 2; (D) the Augustus predicted gene models; (E) the peptides mapping onto Augustus models from Case study 1; (F) the peptides mapping onto the official gene models in release 6 and (G) the official gene models in release 6. Lastly, the 5′ exons that appear to have been missed in release 6 and added in release 10 are boxed (H).

heavily influenced by the overall peptide sequence coverage that is likely to be low for many of these proteins – our analysis presented here is based on a single LC-MS run. However, for the purposes of benchmarking, this demonstrates that ProteoAnnotator is not producing a large number of random matches to non-canonical gene sets. A large-scale meta-analysis of numerous shotgun data sets for *T. gondii* is in progress using ProteoAnnotator, and will be submitted for publication shortly.

As a representative example, we examined the highest scoring *alternative loci* in Case study 1 to demonstrate the effectiveness of the pipeline. The top hit was to Augustus prediction g5397.t1 that was grouped with release 6 gene model TGME49_111470 (Supporting Information File 3). The protein group had a non-canonical gene model score = 476, supported by 20 'non-A' peptides. A search of ToxoDB for TGME49_111470 (release 6 accession) maps this protein to TGME49_311470 (release 10 accession), with functional annotation of 'rhoptry neck protein RON5'. We have uploaded the ProteoAnnotator-produced GFF3 files into ToxoDB, enabling this region to be visualised, as shown in Fig. 3. To produce Fig. 3, a bespoke script was used to re-align the coordinate sets for loci and peptides mapped to release 6 – since the genomic coordinates themselves have altered by a small amount on each chromosome between releases 6 and 10. The results demonstrate that a large number of 5′ exons

had been missed in the release 6 gene prediction (Fig. 3H). These exons were predicted by Augustus and 20 peptides were mapped onto these exons. These exons have now been added to the annotation in release 10, and are supported by high-quality RNASeq data sets. However, it is also interesting to note that the six most 5′ exons appear to have no support in the RNASeq data or from peptides in this study, indicating that perhaps further annotation may yet be required. This example gives a graphical demonstration of the capabilities of ProteoAnnotator for supporting gene model improvements – which, via our historical analysis, we can demonstrate have been independently verified.

In Case study 2, only 15 alternative loci are identified with q-value < 0.05. Many of these are supported by a single-peptide identification – which perhaps may indicate single exons that have been missed in the current annotation, and for those with q values approximately 0.01–0.05 should be treated with caution, since they could be false-positive identifications. However, there are several alternative loci with strong support from a number of alternative peptides, for example Augustus prediction g6646.t1 has a non-canonical gene model score = 392, supported by 12 alternative peptides. Preliminary analysis suggests that this is a predicted transcript from a region of DNA that has not yet been assembled onto a chromosome, and thus missed in the release 10 annotations. This simple example demonstrates that even though the *T. gondii* release 10 gene models are now of high quality, there are still improvements to be made from proteomic data. The data for Case study 1 and 2 have been submitted to ProteomeXchange central under identifiers PXD001042 and DOI 10.6019/PXD001042.

The purpose of Case study 3 was to demonstrate the capability for using ProteoAnnotator with a large input data set, and for mapping against the human genome of considerably higher complexity than the apicomplexan genomes. In the initial publication for data set PXD000652 [26], the authors analysed the data in a search against UniProt – resulting in 1623 protein group identifications (downloaded from the CSF website: 'gel depleted fraction results'). Our analysis searched against the latest Ensembl build yielded 1703 protein groups (Supporting Information File 7). To determine the cross-over in these two sets, we used the Uniprot ID mapping service (http://www.uniprot.org/uploadlists/) to retrieve Ensembl protein identifiers for all of the accessions identified in the CSF study (1799 identifiers within the 1623 groups). The mapping service was able to map 1736 of 1799 accessions (96%) to Ensembl – the reason for the other identifiers not mapping is unknown (but a common problem between different databases). We then performed an analysis to determine how many of the CSF study protein groups had also been identified by our ProteoAnnotator analysis, which was 1531 of 1623 (94%) – see Supporting Information File 7. Given that there is a 4% loss in the identifier mapping process, it appears that there is high agreement between the identifications made in the original study and those made by ProteoAnnotator (approximately >97% agreement). This provides confidence that ProteoAnnotator is functioning correctly to map identifiers to Ensembl and, as demonstrated by the challenges of even mapping identifiers across resources, highlights the importance of having a pipeline that can query mass spectra directly against a genome of interest. Case study 3 has also been uploaded to ProteomeXchange under accession PXD001390 and DOI: 10.6019/PXD001390.

4 Concluding remarks

We present version 1.0 of ProteoAnnotator pipeline, which has been embedded in the MzidLib. We believe the pipeline will assist in proteogenomics studies at various stages of maturity. ProteoAnnotator can be deployed easily via a graphical user interface in our Proteosuite software, or as a command line tool for integration into other projects or for use on high-performance compute clusters. Each module described can be deployed in isolation, fostering re-use and integration into other tool kits. ProteoAnnotator exports file formats for genomic integration (GFF3) and for submission to public proteomics repositories (mzIdentML).

We are continuing to develop ProteoAnnotator – and in the future will release versions capable of functioning on the Galaxy cloud (working with the Bessant Bioinformatics Lab at the Queen Mary University of London) and developing tools for 'blind' modification identification. The design mode also means that new routines, such as improved database design strategies, new search engines, new statistical routines and new visualisation or export formats, can be easily incorporated as new modules within the MzidLib project.

*The MS proteomics data in this paper have been deposited in the ProteomeXchange Consortium (*http://proteomecentral.proteomexchange.org*) via the PRIDE partner repository [18]: dataset identifiers PXD001042 and PXD001390. The authors thank the Toxoplasma Genomic Resource (ToxoDB) for providing the data sets used for the case studies, and the developers of AUGUSTUS and Glimmer gene predication tools. They also thank the Computational Omics and Systems Biology Group, led by Lennart Martens, for the SearchGUI that was re-used in the ProteoAnnotator pipeline. The authors also thank the PRIDE team for assistance in uploading data sets to the public repository. This work was funded by BBSRC grants to A. R. J. [BB/K004123/1, BB/H024654/1] and a BBSRC grant to A. R. J. and J. M. W [BB/G010781/1].*

The authors have declared no conflict of interest.

5 References

[1] Yandell, M., Ence, D., A beginner's guide to eukaryotic genome annotation. *Nat. Rev. Genet.* 2012, *13*, 329–342.

[2] Wang, E. T., Sandberg, R., Luo, S., Khrebtukova, I. et al., Alternative isoform regulation in human tissue transcriptomes. *Nature* 2008, *456*, 470–476.

[3] Perkins, D. N., Pappin, D. J. C., Creasy, D. M., Cottrell, J. S., Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* 1999, *20*, 3551–3567.

[4] Geer, L. Y., Markey, S. P., Kowalak, J. A., Wagner, L. et al., Open mass spectrometry search algorithm. *J. Proteome Res.* 2004, *3*, 958–964.

[5] Craig, R., Cortens, J. P., Beavis, R. C., Open source system for analyzing, validating, and storing protein identification data. *J. Proteome Res.* 2004, *3*, 1234–1242.

[6] Kim, S., Gupta, N., Pevzner, P. A., Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases. *J. Proteome Res.* 2008, *7*, 3354–3363.

[7] Shadforth, I., Xu, W., Crowther, D., Bessant, C., GAPP: a fully automated software for the confident identification of human peptides from tandem mass spectra. *J. Proteome Res.* 2006, *5*, 2849–2852.

[8] Risk, B. A., Spitzer, W. J., Giddings, M. C., Peppy: proteogenomic search software. *J. Proteome Res.* 2013, *12*, 3019–3025.

[9] Pang, C. N. I., Tay, A. P., Aya, C., Twine, N. A. et al., Tools to covisualize and coanalyze proteomic data with genomes and transcriptomes: validation of genes and alternative mRNA splicing. *J. Proteome Res.* 2013, *13*, 84–98.

[10] Goecks, J., Nekrutenko, A., Taylor, J., The Galaxy, T., Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* 2010, *11*, R86.

[11] Kuhring, M., Renard, B. Y., iPiG: integrating peptide spectrum matches into genome browser visualizations. *PLoS One* 2012, *7*, e50246.

[12] Castellana, N. E., Pham, V., Arnott, D., Lill, J. R., Bafna, V., Template proteogenomics: sequencing whole proteins using an imperfect database. *Mol. Cell. Proteomics* 2010, *9*, 1260–1270.

[13] Tanner, S., Shu, H., Frank, A., Wang, L.-C. et al., InSPecT: identification of posttranslationally modified peptides from tandem mass spectra. *Anal. Chem.* 2005, *77*, 4626–4639.

[14] Castellana, N. E., Shen, Z., He, Y., Walley, J. W. et al., An automated proteogenomic method uses mass spectrometry to reveal novel genes in *Zea mays*. *Mol. Cell. Proteomics* 2014, *13*, 157–167.

[15] Jones, A. R., Eisenacher, M., Mayer, G., Kohlbacher, O. et al., The mzIdentML data standard for mass spectrometry-based proteomics results. *Mol. Cell. Proteomics* 2012, *11*, 1–10.

[16] Ghali, F., Krishna, R., Lukasse, P., Martínez-Bartolomé, S. et al., Tools (viewer, library and validator) that facilitate use of the peptide and protein identification standard format, termed mzIdentML. *Mol. Cell. Proteomics* 2013, *12*, 3026–3035.

[17] Vizcaino, J. A., Deutsch, E. W., Wang, R., Csordas, A. et al., ProteomeXchange provides globally coordinated proteomics data submission and dissemination. *Nat. Biotech.* 2014, *32*, 223–226.

[18] Vizcaino, J. A., Cote, R. G., Csordas, A., Dianes, J. A. et al., The PRoteomics IDEntifications (PRIDE) database and associated tools: status in 2013. *Nucleic Acids Res.* 2013, *41*, D1063–D1069.

[19] Vaudel, M., Barsnes, H., Berven, F. S., Sickmann, A., Martens, L., SearchGUI: an open-source graphical user interface for simultaneous OMSSA and X!Tandem searches. *Proteomics* 2011, *11*, 996–999.

[20] Jones, A. R., Siepen, J. A., Hubbard, S. J., Paton, N. W., Improving sensitivity in proteome studies by analysis of false discovery rates for multiple search engines. *Proteomics* 2009, *9*, 1220–1229.

[21] Granholm, V., Navarro, J. F., Noble, W. S., Kall, L., Determining the calibration of confidence estimation procedures for unique peptides in shotgun proteomics. *J. Proteomics* 2012, *80c*, 123–131.

[22] Mayer, G., Montecchi-Palazzi, L., Ovelleiro, D., Jones, A. R. et al., The HUPO Proteomics Standards Initiative – mass spectrometry controlled vocabulary. *Database* 2013, *2013*, bat009.

[23] Kissinger, J. C., Gajria, B., Li, L., Paulsen, I. T., Roos, D. S., ToxoDB: accessing the *Toxoplasma gondii* genome. *Nucleic Acids Res.* 2003, *31*, 234–236.

[24] Stanke, M., Waack, S., Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics* 2003, *19*, ii215–ii225.

[25] Delcher, A. L., Harmon, D., Kasif, S., White, O., Salzberg, S. L., Improved microbial gene identification with GLIMMER. *Nucleic Acids Res.* 1999, *27*, 4636–4641.

[26] Guldbrandsen, A., Vethe, H., Farag, Y., Oveland, E. et al., In-depth characterization of the cerebrospinal fluid proteome displayed through the CSF Proteome Resource (CSF-PR). *Mol. Cell. Proteomics* 2014, M114.038554.

[27] Flicek, P., Amode, M. R., Barrell, D., Beal, K. et al., Ensembl 2014. *Nucleic Acids Res.* 2014, *42*, D749–D755.

