Pout2Prot: an efficient tool to create protein (sub)groups from Percolator output files

Tim Van Den Bossche[1,2#], Kay Schallert[3#], Pieter Verschaffelt[1,4], Bart Mesuere[1,4], Dirk Benndorf[3,5,6], and Lennart Martens[1,2*]

[1]VIB - UGent Center for Medical Biotechnology, VIB, Ghent, Belgium

[2]Department of Biomolecular Medicine, Faculty of Medicine and Health Sciences, Ghent University, Ghent, Belgium

[3]Bioprocess Engineering, Otto-von-Guericke University Magdeburg, Magdeburg, Germany

[4]Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Ghent, Belgium

[5]Microbiology, Department of Applied Biosciences and Process Technology, Anhalt University of Applied Sciences, Köthen, Germany

[6]Bioprocess Engineering, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany

Shared contributions

* Corresponding author: Lennart.Martens@UGent.be

Abstract

The protein inference problem is complicated in metaproteomics due to the presence of homologous proteins from closely related species. Nevertheless, this process is vital to assign taxonomy and functions to identified proteins of microbial species, a task for which specialized tools such as Prophane have been developed. We here present *Pout2Prot*, which takes Percolator Output (.pout) files from multiple experiments and creates protein (sub)group output files (.tsv) that can be used directly with Prophane. *Pout2Prot* offers different grouping strategies, allows distinction between sample categories and replicates for multiple files, and uses a weighted spectral count for protein (sub)groups to reflect (sub)group abundance. *Pout2Prot* is available as a web application at https://pout2prot.ugent.be and is installable via pip as a standalone command line tool and reusable software library. All code is open source under the Apache License 2.0 and is available at https://github.com/compomics/pout2prot.

Keywords: metaproteomics, computational proteomics, protein inference, bioinformatics

Logo:

Introduction

In metaproteomics, the study of the collective proteome of whole (microbial) ecosystems, it is important to learn about the taxonomy and functions represented in the community. For this purpose, tools such as Unipept (Verschaffelt *et al.*, 2021) and Prophane (Schiebenhoefer *et al.*, 2020) have been made available to specifically perform downstream annotation of metaproteomic data, while other, more generic tools also provide connections to downstream annotation tools (Van Den Bossche *et al.*, 2020). These tools, however, work very differently: while Unipept relies on identified peptides without inferring the corresponding proteins (a peptide-centric approach), Prophane only allows protein groups as input (a protein-centric approach). Recently, these two tools were compared in the first multi-lab comparison study in metaproteomics (CAMPI) (Van Den Bossche *et al.*, 2021), which indicated that the choice between these approaches is a matter of user preference.

The process of grouping proteins is unfortunately not as straightforward as it might first appear (Martens and Hermjakob, 2007; Uszkoreit *et al.*, 2015; Audain *et al.*, 2017; Nesvizhskii and Aebersold, 2005). Identified peptide sequences have to be assembled into a list of identified proteins, but when a peptide can be mapped to multiple proteins, this leads to the protein inference problem (Nesvizhskii and Aebersold, 2005). In metaproteomics this problem is exacerbated due to the presence of homologous proteins from multiple species in its necessarily large protein databases (Schiebenhoefer *et al.*, 2019). Protein grouping is therefore commonly used to generate a more manageable list of identified proteins that can be used for further downstream analysis. However, different protein grouping algorithms can be chosen, leading to differing lists of protein groups from a single set of identified peptides (Martens and Hermjakob, 2007). In the past, many protein grouping methods have been developed, as reviewed in Audain et al (Audain *et al.*, 2017), but these typically do not interface well with post-processing tools like Percolator (Käll *et al.*, 2007), which are able to increase the number of peptide-to-spectrum matches (PSMs) due to a better separation of true and false matches (Bouwmeester *et al.*, 2020). Moreover, the common strategy used by these tools is the *Occam's razor* strategy, which is not always ideal in metaproteomics (Van Den Bossche *et al.*, 2021). We here therefore present a new tool, *Pout2Prot*, which provides users with two relevant protein inference options that are tailored towards metaproteomics use case: *Occam's razor* and *anti-Occam's razor* (Martens and Hermjakob, 2007). *Occam's razor* is based on the principle of maximum parsimony and provides the smallest set of proteins that explains all observed peptides. Here, however, proteins that are not matched by a unique peptide are discarded and their associated taxonomy and functions, which might actually be present in the sample, are lost. This algorithm is for example used in the X!TandemPipeline

(Langella *et al.*, 2017). On the other hand, *anti-Occam's razor* is based on the maximal explanatory set of proteins, where any protein that is matched by at least one identified peptide will be included in the reported protein list. This algorithm is used in, for example, MetaProteomeAnalyzer (MPA) (Muth *et al.*, 2015). Unfortunately, there is no simple way to determine a priori which algorithm will be optimal, as this can differ from sample to sample (Van Den Bossche *et al.*, 2021).

Moreover, as proteins are grouped based on their identified peptides, carefully defined rules are required on when and how to group these proteins. There are two possible approaches here: the first approach consists of grouping all proteins that share one or more identified peptides (i.e., the shared peptide rule), while the second approach consists of only grouping proteins that share the same set (or subset) of identified peptides (i.e., the shared peptide set rule). These two approaches can also be interpreted as grouping at two different levels: the protein group level (based on the shared peptide rule) and the protein subgroup level (based on the shared peptide set rule). The two protein grouping strategies (*Occam's razor* and *anti-Occam's razor*) and two approaches (grouping at protein group and subgroup level) are visually represented in **Supplementary Figure 1**.

In this manuscript, we describe a new command line tool and web application that converts .pout files from different files and experiments to a file containing protein (sub)groups that can be used by Prophane. *Pout2Prot* thus enables Percolator users to also use Prophane for downstream functional and taxonomic analysis.

Implementation

Pout2Prot is implemented in Python and installable as a Python package from PyPI. It can then be invoked from the command line. We also provide a user-friendly and easily accessible web application of our tool (https://pout2prot.ugent.be). Protein grouping analysis is efficient and can, consequently, be performed entirely on the user's local machine.

Evaluation

Pout2Prot converts .pout files to protein (sub)group files that can be immediately imported in Prophane for further downstream analysis. This Prophane input file consists of four tab-separated fields: sample category, sample name, protein accessions, and spectrum count. The sample category allows users to divide their experiment in different categories (e.g. "control" and "disease"). If no sample categories are provided, these values will be identical to the sample name, which results in individual quantification by Prophane. The sample name is

identical to the name of the .pout file, so each protein (sub)group can be traced back to its origin file. The protein accessions will contain the proteins present in the protein (sub)group, based on the chosen strategy. Finally, the spectrum count contains the weighted spectrum count from all PSMs present in that protein (sub)group, with PSMs present in multiple subgroups counted as fractional values in each subgroup.

To test *Pout2Prot*, we constructed several toy examples. These are available on GitHub and are explained in **Supplementary File 1**. Furthermore, we tested *Pout2Prot* on a metaproteomics dataset (results in **Supplementary File 1**).

Conclusion

Pout2Prot enables the conversion of Percolator output (.pout) files to protein group and protein subgroup files, based on either the *Occam's razor* or *anti-Occam's razor* strategy, and therefore closes an important gap in the bioinformatic workflow of metaproteomics data analysis. *Pout2Prot* also allows the user to create protein (sub)groups across experiments and across sample categories. The output of *Pout2Prot* can be imported directly into Prophane, which in turn allows users to perform downstream taxonomic and functional analysis of metaproteomics samples.

Acknowledgements

This work has benefited from collaborations facilitated by the Metaproteomics Initiative (https://metaproteomics.org/) whose goals are to promote, improve and standardize metaproteomics.

Funding

TVDB, PV, LM and BM would like to acknowledge the Research Foundation - Flanders (FWO) [grants 1S90918N, 1164420N, G042518N and 12I5220N]. LM also acknowledges support from the European Union's Horizon 2020 Programme under Grant Agreement 823839 [H2020-INFRAIA-2018-1]. KS and DB would like to acknowledge the German Federal Ministry of Education and Research (BMBF) of the project 'MetaProteomanalyzer Service' within the German Network for Bioinformatics Infrastructure (de.NBI) [031L103]. The authors declare no conflict of interest.

References

Audain,E. *et al.* (2017) In-depth analysis of protein inference algorithms using multiple search engines and well-defined metrics. *J. Proteomics*, **150**, 170–182.

Bouwmeester,R. *et al.* (2020) The Age of Data-Driven Proteomics: How Machine Learning Enables Novel Workflows. *Proteomics*, e1900351.

Käll,L. *et al.* (2007) Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat. Methods*, **4**, 923–925.

Langella,O. *et al.* (2017) X!TandemPipeline: A Tool to Manage Sequence Redundancy for Protein Inference and Phosphosite Identification. *J. Proteome Res.*, **16**, 494–503.

Martens,L. and Hermjakob,H. (2007) Proteomics data validation: why all must provide data. *Mol. Biosyst.*, **3**, 518–522.

Muth,T. *et al.* (2015) The MetaProteomeAnalyzer: a powerful open-source software suite for metaproteomics data analysis and interpretation. *J. Proteome Res.*, **14**, 1557–1565.

Nesvizhskii,A.I. and Aebersold,R. (2005) Interpretation of shotgun proteomic data: the protein inference problem. *Mol. Cell. Proteomics*, **4**, 1419–1440.

Schiebenhoefer,H. *et al.* (2020) A complete and flexible workflow for metaproteomics data analysis based on MetaProteomeAnalyzer and Prophane. *Nat. Protoc.*, **15**, 3212–3239.

Schiebenhoefer,H. *et al.* (2019) Challenges and promise at the interface of metaproteomics and genomics: an overview of recent progress in metaproteogenomic data analysis. *Expert Rev. Proteomics*, **16**, 375–390.

Uszkoreit,J. *et al.* (2015) PIA: An Intuitive Protein Inference Engine with a Web-Based User Interface. *J. Proteome Res.*, **14**, 2988–2997.

Van Den Bossche,T. *et al.* (2020) Connecting MetaProteomeAnalyzer and PeptideShaker to Unipept for Seamless End-to-End Metaproteomics Data Analysis. *J. Proteome Res.*, **19**, 3562–3566.

Van Den Bossche,T. *et al.* (2021) Critical assessment of metaproteome investigation (CAMPI): A multi-lab comparison of established workflows. *bioRxiv*.

Verschaffelt,P. *et al.* (2021) Unipept Desktop: A Faster, More Powerful Metaproteomics Results Analysis Tool. *J. Proteome Res.*, **20**, 2005–2009.