Journal of proteome research

Technical Note

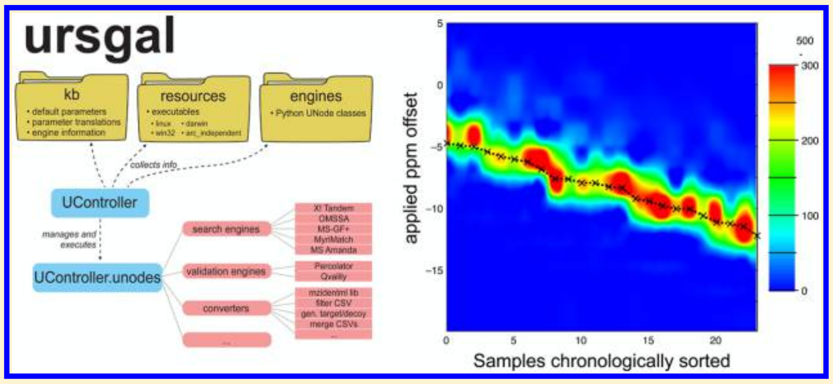
pubs.acs.org/jpr

Ursgal, Universal Python Module Combining Common Bottom-Up Proteomics Tools for Large-Scale Analysis

Lukas P. M. Kremer, Johannes Leufken, Purevdulam Oyunchimeg, Stefan Schulze, and Christian Fufezan*

Institute of Plant Biology and Biotechnology, University of Muenster, Schlossplatz 8, 48143 Münster, Germany

Supporting Information



ABSTRACT: Proteomics data integration has become a broad field with a variety of programs offering innovative algorithms to analyze increasing amounts of data. Unfortunately, this software diversity leads to many problems as soon as the data is analyzed using more than one algorithm for the same task. Although it was shown that the combination of multiple peptide identification algorithms yields more robust results,[1−3] it is only recently that unified approaches are emerging;[4,5] however, workflows that, for example, aim to optimize search parameters or that employ cascaded style searches[6] can only be made accessible if data analysis becomes not only unified but also and most importantly scriptable. Here we introduce Ursgal, a Python interface to many commonly used bottom-up proteomics tools and to additional auxiliary programs. Complex workflows can thus be composed using the Python scripting language using a few lines of code. Ursgal is easily extensible, and we have made several database search engines (X!Tandem,[7] OMSSA,[8] MS-GF+,[9] Myrimatch,[10] MS Amanda[11]), statistical postprocessing algorithms (qvality,[12] Percolator[13]), and one algorithm that combines statistically postprocessed outputs from multiple search engines ("combined FDR"[14]) accessible as an interface in Python. Furthermore, we have implemented a new algorithm ("combined PEP") that combines multiple search engines employing elements of "combined FDR",[14] PeptideShaker,[2] and Bayes' theorem.

KEYWORDS: *Python, peptide identification, search engine, statistical postprocessing engine, high throughput, combine search engine results*

1. INTRODUCTION

Mass-spectrometry-based bottom-up proteomic studies require robust peptide identifications. Without this, all ensuing analyses and biological interpretations suffer substantially. Up to now, three strategies have been developed to link the measured mass spectrometry spectra to the peptide identities (termed PSMs for peptide spectrum matches) using (a) databases build on theoretical fragmentation spectra purely devised from sequence information, (b) peptide spectral library search, or (c) de novo algorithms, but it is left up to researchers to decide which strategy is best suited to their project. Naturally, the answer is all of them, yet universal approaches are still missing that combine the different methods due to the lack of a unified access to specialized tools. The most common strategy is database search, given that it offers a reliable calculation of false discovery rates (FDRs) or posterior error probabilities (PEPs) for PSMs. Several statistical approaches have been developed for this type of calculation, increasing the number of ways proteomic data can be analyzed. The combination of multiple search algorithms increases sensitivity and reliability,[4,3,14,15,1] yet a unified approach that would allow many different proteomic data evaluations through a simple scriptable programming language is not available. The development of tools like SearchGUI,[4] PeptideShaker,[2] DeNovoGUI,[16] TOPAS,[17] Galaxy,[18] KNIME,[19] TPP,[20] and Proteomatic[21] was a first step toward simplifying different approaches in a very user-friendly way; however, they are focused on linear workflows and thus they are less suitable for iterative scriptable approaches. For example, the reanalysis of large-scale experiments, which helps to optimize parameters and thus ensures the highest sensitivity or robustness[22] should, from a

Special Issue: Large-Scale Computational Mass Spectrometry and Multi-Omics

Received: September 14, 2015
Published: December 28, 2015

programming point of view, be easy to incorporate. A few lines of code such as

```
>>> for mzML in all_new_mzMLs:
>>>  for machine_offset in ['10ppm', '50ppm', '100ppm', '500ppm']:
>>>    mgf = mzml2mgf( mzML, offset=machine_offset )
>>>    for search_engine in ['xtandem', 'omssa', 'msgf', 'myrimatch', 'msamanda']:
>>>       results = search( mgf, engine=search_engine )
```

seem obvious, yet such an implementation remains difficult because (a) a simple mzML to mgf converter that recalibrates mgf files cannot be easily chained to a workflow, (b) a generalized, for example, search function is not available due to the lack of unified parameters, and, most importantly, (c) no scriptable interface to search or statistical postprocessing engines is available. Another example is a cascaded search approach as used by Kertesz-Farkas et al.[6] to analyze unassigned spectra with modified parameters or additional post-translational modifications. Again, from a programming point of view, such an implementation into established analysis workflows seems straightforward; however, a unified scriptable access to analysis tools is lacking, making it more complex than it should be.

Here we introduce such an interface called Ursgal (Mongolian: flow), which is open source and publicly available on GitHub (https://github.com/ursgal/ursgal). Our idea and philosophy behind Ursgal is to make computational proteomics unified and scriptable, offering the possibility to exchange or extend any processing steps and to employ complex nested loops as shown above. This design simplifies the incorporation of new ideas or parameter fine-tuning into established workflows. For example, a researcher can incorporate a novel target-decoy database generation approach with ease into a fully fledged workflow, containing, for example, multiple search and statistical postprocessing engines. Ursgal is written in pure Python code and is therefore truly platform-independent (OS X, Linux, and Windows), quickly deployable, extendible and most importantly, the source code was written with an emphasis on readability. Of course, many of the standalone extensions that are wrapped by Ursgal are not written in Python and those are not necessarily platform-independent. Thus, besides Python 3.4 or higher and pymzML,[23] Ursgal's dependencies are defined by the incorporated standalone programs that require, for example, Java (https://www.java.com/en/). The standalone programs in Ursgal are called UNodes. Each UNode is defined by a file or folder in the "resources", "kb" (knowledge base), and "engines" directories (Figure 1A; for more details please refer to the online documentation at http://ursgal.readthedocs.org/). The UController collects and validates all UNode information and is used as the starting point to run the UNodes. The UController offers wrapper functions, such as UController.search() or UController.validate(), or can run UNodes directly via UController.execute_unode() (Figure 1B). The UController uses JSON formatted files to communicate the parameters and history of the input file to the UNodes. This allows any workflow to be stopped and later continued on any other machine running Ursgal without further information by simply supplying the input file and its JSON. This design offers the possibility to use established queuing systems for grid infrastructures, such as HTCondor,[24] or to share parameters with other researchers so results can be reproduced. Furthermore, the modularity allows different tasks to be executed on appropriate nodes. For example, memory-intensive tasks, such as mzIdentML conversions (using the mzIdentML library[25]), can be scheduled on nodes with much random access memory (RAM) or tasks that do not dig into the computing

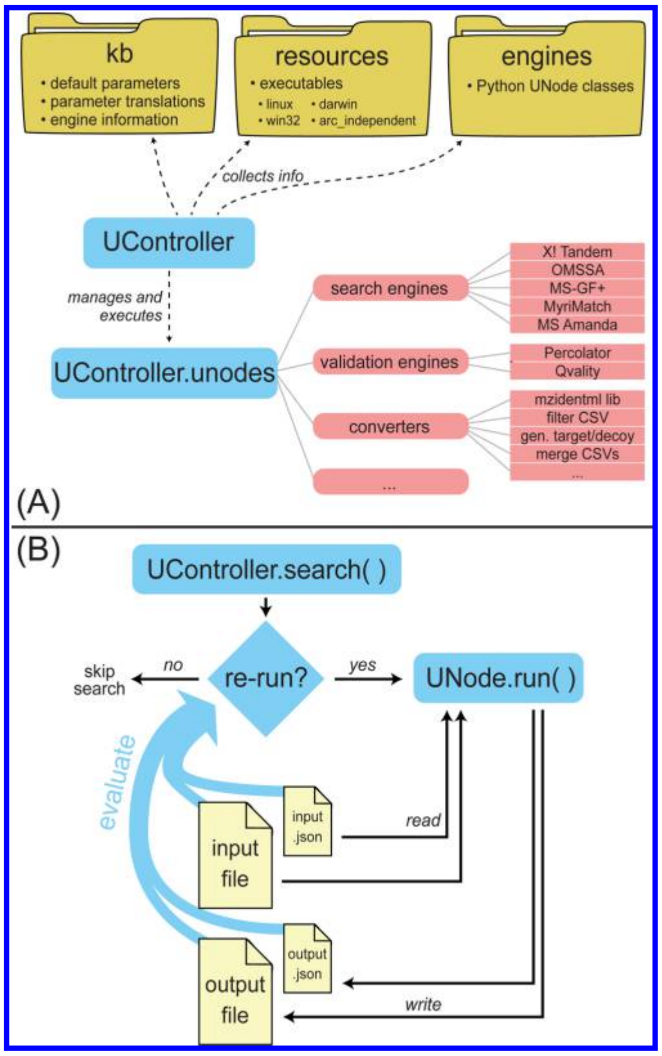


Figure 1. Schematic overview of Ursgal file objects (a) and general UNode execution flow (b).

power of modern CPUs; that is, single thread applications (e.g., mzml2mgf.py) can be executed multiple times on nodes with multiple cores.

Up to now we have incorporated established programs such as MS Amanda (v. 1.0.0.5242 and 1.0.0.5243[11]), MS-GF+ (v. 9979[9]), Myrimatch (v. 2.1.138 and 2.1.140[10]), OMSSA (v. 2.1.9[8]), Percolator (v. 2.0.6 and 2.0.8[13]), qvality (v. 2.02[12]), and X!Tandem (v. Cyclone.2010, Jackhammer, Sledgehammer, and Piledriver[26]). Additionally, we have incorporated the mzIdentML library (v. 1.6.10 and 1.6.11[25]), which offers the possibility to convert between different standardized formats, for example, from mzID or XML to a more readable format, that is, CSV. Furthermore, we have implemented an established method ("combined FDR"[14]) and a new method, "combined PEP", that merges results from different database search engines. Our new approach "combined PEP" joins elements of PeptideShaker,[2] the combined FDR algorithm,[14] and Bayes' theorem. Also, we have included several auxiliary programs that are useful for computational biologists, such as a target-decoy generating engine or HTTP and FTP download nodes. Thus, in total, Ursgal gives unified scriptable access to (depending on the system architecture) five database search engines, two statistical postprocessing engines, and two engines that can combine the results from multiple database search engines.

Novel engines can be incorporated using a simple text editor within a few minutes following simple rules, which are described in detail in the online documentation (http://ursgal.readthedocs.org/). Because the mzIdentML library can convert mzID to CSV, any new engine supporting mzID can be incorporated easily into Ursgal workflows.

2. METHODS

2.1. Parameter Optimization

The data set by Bruderer et al.[27] (measured in data dependent acquisition (DDA) mode) was used to show an application of a parameter optimization approach using Ursgal. RAW mass spectrometry data was obtained from peptideatlas.org (PASS00589) and converted to mzML using msconvert from Proteowizard (v. 3.0.7408).[28] The reported MaxQuant (v. 1.4.1.2) peptide identification results were analyzed and the "Uncalibrated Mass Error [ppm]" for each PSM was grouped on file level. The global machine offset on MS run/file level was then calculated by estimating the maximum of the density distribution described by all reported PSMs and their "Uncalibrated Mass Error [ppm]" (Figure S1). We then used Ursgal to reproduce the observed ppm error reported by MaxQuant and to test whether the precursor and fragment mass tolerance parameters would have an influence on the ppm error estimation. For this, a nested loop was designed that sweeps (a) the precursor mass tolerance from 1 to 5 ppm in 1 ppm steps, (b) the fragment mass tolerance from 2.5 to 20 ppm in four steps (2.5, 5, 10, and 20 ppm), and (c) the machine offset from −20 to 20 ppm in 2 ppm steps, which is used to recalibrate the m/z values during conversion from mzML to mgf files using the mzml2mgf engine in Ursgal. To speed up the analysis, we used just every tenth MS^2 spectrum. Carbamidomethylation was defined as fixed modification. Overall, this resulted in a total of 420 searches per mzML file (total 10 080 searches on 24 mzML files) that were performed using X!Tandem (v. Piledriver). Statistical postprocessing and calculation of PEPs were done using Percolator (v. 2.08). The results were filtered by PEP $\leq$ 1%, and all decoys were removed. For each parameter combination in our sweep (a: precursor mass tolerance, b: fragment mass tolerance, and c: machine offset value), the numbers of accepted and unique peptides were counted.

2.2. Combining Multiengine Search Results with "Combined PEP"

"Combined PEP" is a hybrid approach combining elements of the "combined FDR" approach,[14] elements of PeptideShaker,[2] and Bayes' theorem. Similar to "combined FDR", "combined PEP" groups the PSMs. For each search engine, the reported PSMs are treated as a set and the logical combinations of all sets are treated separately as done in the "combined FDR" approach. For instance, three search engines would result in seven PSM groups, which can be visualized by the seven intersections of a three-set Venn diagram. Typically, a PSM group that is shared by multiple engines contains fewer decoy hits and thus represents a higher quality subset and thus its PSMs receive a higher score. This approach is based on the assumption that the search engines agree on the decoys and false-positives as they agree on the targets. While this is still a matter of debate, a more detailed analysis would go beyond this work and thus has to be addressed elsewhere. As with the other examples of this work, our aim with Ursgal was to show how easily such approaches are implemented.

The combined PEP approach uses Bayes' theorem to calculate a multiengine PEP (MEP) for each PSM based on the PEPs reported by, for example, Percolator for different search engines, that is

$$MEP = (PEP_1 \times PEP_2 \times ... \times PEP_N)/(PEP_1 \times PEP_2 \times ... \times PEP_N + (1 - PEP_1) \times (1 - PEP_2) \times ... \times (1 - PEP_N))$$

This is done for each PSM group separately.

Then, the combined PEP (the final score) is computed similar to PeptideShaker[2] using a sliding window over all PSMs within each group (sorted by MEP). Each PSM receives a PEP based on the target/decoy ratio of the surrounding PSMs.

$$\text{combined } PEP = \frac{2 \times \text{number of decoys in window}}{\text{number of total PSMs in window}}$$

Finally, all groups are merged and the results reported in one output, including all the search result scores from the individual search engines as well as the FDR based on the "combined PEP".

The combined PEP analysis of the ROS experiment was conducted using the database, search parameters, and modifications as previously described.[29] The data were obtained by three experiments, each of which contained four biological conditions. Each protein sample was digested by FASP,[30] and resulting peptides were fractionated using Post-FASP,[30] yielding six LC−MS measurements and thus 72 mzML files in total. For each of the 12 data sets, the search results of all Post-FASP pH-fractions were merged and then statistically postprocessed with Percolator (v. 2.08). Figure 3 shows the numbers of accepted peptides (unique combinations of sequence and modification) for FDR thresholds ranging from 0 to 5%.

A complete Ursgal workflow for combined PEP and combined FDR score was tested on the data set Human-BR.[31] The same data set was recently used by Wen et al.[5] and consists of four RAW files (120813OTc1_NQL-AU-0314-LFQ-LCM-SG-*.RAW, ProteomeXchange Consortium, accession PXD000278). RAW files were converted to mzML format using msconvert from ProteoWizard (v. 3.0.8738).[28] The data are part of a proteomics analysis of breast cancer tissues. Detailed information on data generation methods can be found in Liu et al.[31]

All mzML files were searched for peptides using MS Amanda (v. 1.0.0.5242), MyriMatch (v. 2.1.1.138), MS-GF+ (v. 9976), OMSSA (v. 2.1.9), and X!Tandem (v. Sledgehammer). Search engine parameters and modifications were used as described in Wen et al.[5] Because the authors' exact database was not available online, we downloaded all proteins of the UniProt Human Reference Proteome (UP000005640) from the same download date. PEPs and q-values were calculated using Percolator (v. 2.08) and then merged into a single result file for each search engine. Postprocessed search results were then combined using Ursgal's "combined FDR" and "combined PEP" UNodes.

2.3. Cascade Search

The Human-BR data set was used for a cascade search approach similar to Kertesz-Farkas et al.[6] A series of searches for different types of modifications was conducted: (1) no modification, (2) oxidation of M, (3) deamidation of N/Q, (4) methylation of E/K/R, (5) N-terminal acetylation, and (6) phosphorylation of S/T. After each step, the PEP was estimated using qvality (v. 2.02), and all spectra with a target PSM below

1% PEP were removed. In contrast to Kertesz-Farkas et al.,[6] the PEP was estimated based on a target-decoy database (*Homo sapiens* target database of UniProt/SwissProt, download date September 7, 2015, decoy generation by tryptic peptide shuffling), and all searches were conducted against the same database. For comparison, a search for all modifications at once was conducted, and the results were statistically postprocessed with qvality either for the union of PSMs (ungrouped approach) or for each type of modification separately (grouped approach). For each approach, four search engines, (X!Tandem (v. Piledriver), OMSSA (v. 2.1.9), MS-GF+ (v. 9979), and MS Amanda (v. 1.0.0.5243)) were used with Ursgal default parameters using the profile "LTQ XL low res" (precursor ion mass tolerance: 5 ppm, fragment ion mass tolerance: 0.5 Da) and carbamidomethylation of C as fixed modification. The results of all engines were combined. Spectra with multiple PSMs were sanitized based on PEP; that is, only the best hit (differing from the second best hit by 2 orders of magnitude) for one spectrum was accepted. The identified peptides (combination of peptide sequence and modifications) and PSMs (peptide sequence, modifications, and spectrum ID) were counted. The same procedure was employed for a subset of the Barth et al. data set[29] (Supplemental Table S1).

3. RESULTS

A common belief in proteomics data analysis is that some engines perform better than others. Unfortunately, this misconception sometimes originates from the fact that some programs can be used optimally without much user input, while others require in-depth knowledge and fine-tuning.[32] Additionally, the identity of the biological sample and the type of measurement can influence the performance of database search engines. Not to be mislead by such pitfalls, it is good practice to use multiple approaches in general;[1−4,15] however, to analyze proteomic data in depth one requires a scriptable interface that offers a unified set of parameters. Ursgal offers such functionality.

In the following paragraphs we show three examples of how scriptable and unified approaches can improve data evaluation: (a) a parameter optimization (~100 lines of code), (b) a cascade search (~300 lines of code), and (c) a large-scale analysis with five engines and "combined PEP" calculation (~200 lines of code).

3.1. Parameter Optimization

Mass spectrometry machines require regular calibration, which has to be performed at regular intervals and depending on the machine type and environment, more or less often. If not calibrated, the reported m/z values will shift over time. Not all analysis tools can cope with such a drift, and recently Gibbons et al.[33] presented a way to correct such drifts. We evaluated such a drift using the mass spectrometry data published by Bruderer et al.[27] The authors used MaxQuant[34] for peptide identification, which compensates for the drift and reports the observed machine offset as "Uncalibrated Mass Error [ppm]" (Supplementary Figure S1). We used Ursgal using a parameter optimization approach to determine the machine offset in ppm by iteratively adjusting the m/z values in the 24 mgf files for offsets ranging from −20 to +20 ppm in 2 ppm steps and running a search with X!Tandem. We additionally adjusted the precursor mass tolerance from 1 to 5 ppm in 1 ppm steps and the fragment mass tolerance from 2.5 to 20 ppm in 4 steps (2.5, 5, 10, and 20 ppm) to evaluate if those parameter have an influence on the results. Overall, this approach resulted in a total of 10 080 X!Tandem searches, followed by statistical postprocessing with Percolator. Figure 2 shows a heat map of the results obtained using precursor and fragment ion tolerance of 1 and 5 ppm, respectively. The x-axis shows the measured files in chronological order, as reported by Bruderer et al.;[27] the y-axis shows the ppm shift used to recalibrate the mgf and the heat represents the number of unique peptides with a PEP ≤ 1%. The drift of the machine is clearly observable and mirrors the "Uncalibrated Mass Error [ppm]" that was reported by MaxQuant (Bruderer et al.[27] and black dashed line in Figure 2). Thus, by using Ursgal and such a parameter optimization one can easily determine the machine offset and recalibrate the m/z values in the mgf files accordingly. Importantly, this m/z value recalibration allows other database search engines to search mass spectrometer data that inherently have a machine offset, an often neglected fact and reason why some results cannot be reproduced by some search engines.

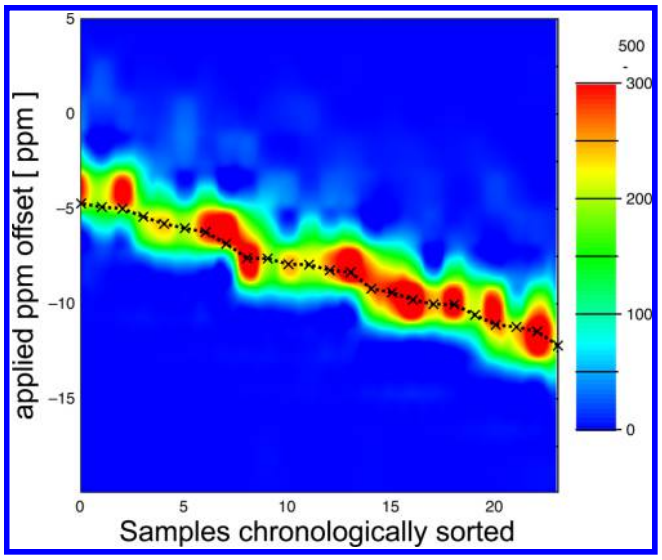


Figure 2. Parameter optimization to determine machine offset in ppm for 24 consecutive measured files (x axis). Y axis shows the applied ppm offset from −20 to +5 ppm in 2 ppm steps (zoom-in from a total range from −20 to +20 ppm). These ppm offsets were used to recalibrate the mgf files prior database search. The heat reflects the peptide count (PEP ≤ 1%) in the respective sample for a given ppm offset bin. Missing values were set to 0. The results obtained with precursor and fragment ion tolerance of 1 and 5 ppm, respectively, are shown, representing a subset of 480 of the total 10 080 performed searches. The black dashed line represents the "Uncalibrated Mass Error [ppm]" that was reported by Bruderer et al.[27] using MaxQuant[34]

3.2. Cascade Search

As a second example, we performed a cascaded search similar to Kertesz-Farkas et al.[6] The Human-BR data set[5,31] was analyzed in consecutive searches for peptides with no modification, oxidation of M, deamidation of N/Q, methylation of E/K/R, N-terminal acetylation, and phosphorylation of S/T. After each step, spectra with target PSMs (PEP ≤ 1%) were removed, thereby reducing the number of spectra in each step while changing the search space over the whole cascade. In total, this resulted in more identified peptides compared with the ungrouped (PEP estimation performed on all PSMs) and grouped (PEP estimation performed for each type of modification separately) approach (Table 1). An increase in the number of unmodified peptides mainly contributed to this

Table 1. Number of Identified Peptides and PSMs at 1% PEP Using a Cascade Search Approach[a]

approach	unmodified	oxidized	deamidated	methylated	acetylated	phosphorylated	multimodified	total
				identified peptides				
ungrouped	6840	150	167	28	138	59	39	7421
grouped	7535	158	101	30	179	59	0	8062
cascade	7737	170	144	38	170	53	0	8312
				PSMs				
ungrouped	19026	346	319	97	342	120	51	20301
grouped	21461	380	164	100	416	139	0	22660
cascade	19299	417	239	99	416	137	0	20607

[a]Human-BR dataset[31] was searched for peptides using a cascade search approach similar to Kertesz-Farkas et al.[6] for which spectra were first searched for unmodified peptides, followed by consecutive searches for the following modifications: oxidation of M, deamidation of N/Q, methylation of E/K/R, N-terminal acetylation, and phosphorylation of S/T. After each step, spectra with a PSM below 1% PEP were removed. This approach was compared to a search for all modifications at once, for which the PEP was estimated either for the union of PSMs (ungrouped) or for each type of modification separately (grouped). Searches were conducted with four search engines (X!Tandem, OMSSA, MS-GF+, and MS Amanda).

higher total peptide count, while for some modifications a decreased number of peptides was observed; however, the highest number of PSMs was identified in the grouped approach. This is in contrast to Kertesz-Farkas et al.[6] and can probably be explained by the fact that we estimated the PEP based on a target-decoy approach and used the same database for all searches. Kertesz-Farkas et al.[6] controlled the FDR using exact *p*-values, which resulted in a target-decoy FDR below 1%, especially for PSMs without peptide modification and finally in a lower number of PSMs. The same trend was observed analyzing the reactive oxygen species (ROS) data set from Barth et al.[29] (Supplemental Table S1). Thus, this example illustrates the easy and suitable implementation of complex workflows like a cascaded search using Ursgal.

3.3. Combination of Search Results from Multiple Engines

The combination of several search engines is not a trivial task and different methodologies currently exist.[1−3,14,15] We have reimplemented the "combined FDR" approach in Python and included it in Ursgal. Additionally, we present here another way to combine search results of multiple engines called "combined PEP". "Combined PEP" joins elements of the "combined FDR" approach described by Jones et al.,[5,14] PeptideShaker described by Vaudel et al.,[2] and Bayes' theorem. To compare our implementation of both methods, we repeated the analysis of Wen et al.[5] (Supplementary Figure S2) and the analysis of the larger dataset by Barth et al.[29] The latter contained 72 mzML files, which were searched, statistically postprocessed, merged, and filtered by FDR. MS Amanda, MyriMatch, MS-GF+, OMSSA, and X!Tandem identified between 5410 and 7037 unique peptides at an FDR cutoff of 1% (Figure 3). Combining these search results with the "combined PEP" or "combined FDR" method yielded over 8000 peptides in both cases, which strengthens the idea that using multiple engines can improve results.

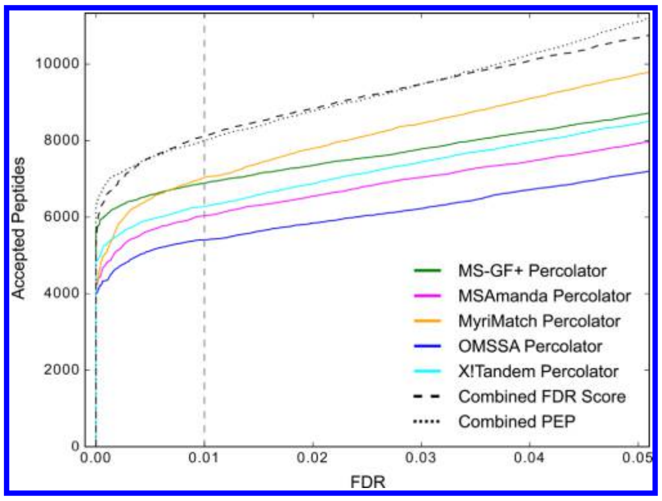


Figure 3. Comparison of five database search engines and their combined results. All search results were statistically postprocessed with Percolator: MS Amanda (violet), MS-GF+ (green), MyriMatch (orange), OMSSA (blue), and X!Tandem (cyan). Search results of all five engines were integrated with the "combined PEP" (black dotted)/ "combined FDR" (black dashed) approach.

4. DISCUSSION

Here we introduce Ursgal, a unified scriptable interface from Python to many commonly used proteomics data analysis tools. Connecting proteomics data analysis and integration to Python offers many opportunities because Python is a well-established open-source programming language that has a large scientific code base. Thus, researchers can extend their analysis with many established packages. Ursgal offers the possibility to perform complex multidimensional data analysis in terms of parameter variation, using multiple database search engines and statistical postprocessing engines with fewer than 100 lines of Python code. This type of analysis will allow researchers to optimize their workflow and mass spectrometry setup and thus potentially offer a deeper insight into their biological questions. Overall, Ursgal allows the rapid development of novel workflows that require scriptable and unified access to mass spectrometry analysis tools. Especially nested loops over, for example, many data sets in complex folder structures or parameter sets, that trigger thousands of searches are not possible in current linear pipeline systems. Thus, such flexible access allows large-scale workflows to be deployed reproducibly with ease. Future development of Ursgal will aim to implement further proteomic data analysis approaches, for example, de novo searches and peptide quantification (pyQms (manuscript in prep.)) or state-of-the-art approaches for the calculation of peptide and protein FDRs like iProphet.[35]

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jproteome.5b00860. Ursgal is freely available at GitHub via https://

github.com/ursgal/ursgal. Supported systems are all featuring Python 3.4 or higher. Several programs/libraries/algorithms may require a specific platform (Windows, Unix, Mac) and/or architecture (32/64bit) or additional software to be installed (e.g., .NET (http://www.microsoft.com/net), Java (https://www.java.com/en/), or Mono (http://www.mono-project.com/)). Please refer to the according documentations for further information. Ursgal is published under MIT license; please refer to LICENSE.txt in the GitHub repository. The full documentation can be found at http://ursgal.readthedocs.org/en/latest/ or can be built from the git repository. For test data used in this publication please refer to the original publications: Parameter optimization was performed on data from Bruderer et al.,[36] and cascade search and combined FDR approach were performed on data from Liu et al.[31] and on a data subset from Barth et al.[29]

Supplementary Table S1: Number of identified peptides and PSMs at 1 % PEP using a cascade search approach. (PDF)

Supplementary Figure S1. Density plot for "Uncalibrated Mass Error [ppm]" of the peptide identification results from Bruderer et al. (2015). Supplementary Figure S2. Comparison of five database search engines and their combined results. (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: fufezan@uni-muenster.de.

Author Contributions

L.P.M.K., J.L., P.O., and S.S. contributed equally to this work

Author Contributions

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The support and feedback of Prof. Dr. M. Hippler, his team, Dr. L. Malmström, and Dr. M. Vaudel is gratefully acknowledged. S.S. acknowledges support from the Deutsche Forschungsgemeinschaft (HI 739/13-1).

■ REFERENCES

(1) Nahnsen, S.; Bertsch, A.; Rahnenführer, J.; Nordheim, A.; Kohlbacher, O. Probabilistic consensus scoring improves tandem mass spectrometry peptide identification. *J. Proteome res.* **2011**, *10* (8), 3332–3343.

(2) Vaudel, M.; Burkhart, J. M.; Zahedi, R. P.; Oveland, E.; Berven, F. S.; Sickmann, A.; Martens, L.; Barsnes, H. PeptideShaker enables reanalysis of MS-derived proteomics data sets. *Nat. Biotechnol.* **2015**, *33* (1), 22–24.

(3) Kwon, T.; Choi, H.; Vogel, C.; Nesvizhskii, A. I.; Marcotte, E. M. MSblender: A probabilistic approach for integrating peptide identifications from multiple database search engines. *J. Proteome res.* **2011**, *10* (7), 2949–2958.

(4) 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* (5), 996–999.

(5) Wen, B.; Du, C.; Li, G.; Ghali, F.; Jones, A. R.; Käll, L.; Xu, S.; Zhou, R.; Ren, Z.; Feng, Q. IPeak: An open source tool to combine results from multiple MS/MS search engines. *Proteomics* **2015**, *15*, 2916.

(6) Kertesz-Farkas, A.; Keich, U.; Noble, W. S. Tandem Mass Spectrum Identification via Cascaded Search. *J. Proteome Res.* **2015**, *14*, 3027.

(7) Craig, R.; Beavis, R. C. A method for reducing the time required to match protein sequences with tandem mass spectra. *Rapid Commun. Mass Spectrom.* **2003**, *17* (20), 2310–2316.

(8) Geer, L. Y.; Markey, S. P.; Kowalak, J. A.; Wagner, L.; Xu, M.; Maynard, D. M.; Yang, X.; Shi, W.; Bryant, S. H. Open Mass Spectrometry Search Algorithm. *J. Proteome res.* **2004**, 3 (5), 958–964.

(9) Kim, S.; Mischerikow, N.; Bandeira, N.; Navarro, J. D.; Wich, L.; Mohammed, S.; Heck, A. J. R.; Pevzner, P. a. The Generating Function of CID, ETD, and CID/ETD Pairs of Tandem Mass Spectra: Applications to Database Search *. *Mol. Cell. Proteomics* **2010**, *9*, 2840–2852.

(10) Tabb, D. L.; Fernando, C. G.; Chambers, M. C. MyriMatch:highly accurate tandem mass spectral peptide identificaiton by multivariate hypergeometric analysis. *J. Proteome Res.* **2007**, *6* (2), 654–661.

(11) Dorfer, V.; Pichler, P.; Stranzl, T.; Stadlmann, J.; Taus, T.; Winkler, S.; Mechtler, K. MS Amanda, a Universal Identification Algorithm Optimised for High Accuracy Tandem Mass Spectra. *J. Proteome res.* **2014**, *13*, 3679.

(12) Käll, L.; Storey, J. D.; Noble, W. S. Qvality: Non-parametric estimation of q-values and posterior error probabilities. *Bioinformatics* **2009**, *25* (7), 964–966.

(13) Käll, L.; Canterbury, J. D.; Weston, J.; Noble, W. S.; MacCoss, M. J. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat. Methods* **2007**, *4* (11), 923–925.

(14) 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* (5), 1220–1229.

(15) Uszkoreit, J.; Maerkens, A.; Perez-Riverol, Y.; Meyer, H. E.; Marcus, K.; Stephan, C.; Kohlbacher, O.; Eisenacher, M. PIA: An Intuitive Protein Inference Engine with a Web-Based User Interface. *J. Proteome res.* **2015**, *14* (7), 2988–2997.

(16) Muth, T.; Weilnböck, L.; Rapp, E.; Huber, C. G.; Martens, L.; Vaudel, M.; Barsnes, H. DeNovoGUI: An open source graphical user interface for de novo sequencing of tandem mass spectra. *J. Proteome res.* **2014**, *13* (2), 1143–1146.

(17) Junker, J.; Bielow, C.; Bertsch, A.; Sturm, M.; Reinert, K.; Kohlbacher, O. TOPPAS: A graphical workflow editor for the analysis of high-throughput proteomics data. *J. Proteome res.* **2012**, *11*, 3914–3920.

(18) Giardine, B.; Riemer, C.; Hardison, R. C.; Burhans, R.; Elnitski, L.; Shah, P.; Zhang, Y.; Blankenberg, D.; Albert, I.; Taylor, J.; et al. Galaxy: A platform for interactive large-scale genome analysis. *Genome Res.* **2005**, *15* (10), 1451–1455.

(19) Berthold, M.; Cebron, N.; Dill, F. KNIME: The Konstanz information miner. *Data Analysis, Machine Learning and Applications* **2008**, *11* (1), 26–31.

(20) Keller, A.; Eng, J.; Zhang, N.; Li, X.; Aebersold, R. A uniform proteomics MS/MS analysis platform utilizing open XML file formats. *Mol. Syst. Biol.* **2005**, *1*, E1–E8.

(21) Specht, M.; Kuhlgert, S.; Fufezan, C.; Hippler, M. Proteomics to go: Proteomatic enables the user-friendly creation of versatile MS/MS data evaluation workflows. *Bioinformatics* **2011**, *27* (8), 1183–1184.

(22) Chick, J. M.; Kolippakkam, D.; Nusinow, D. P.; Zhai, B.; Rad, R.; Huttlin, E. L.; Gygi, S. P. A mass-tolerant database search identifies a large proportion of unassigned spectra in shotgun proteomics as modified peptides. *Nat. Biotechnol.* **2015**, *33* (7), 743.

(23) Bald, T.; Barth, J.; Niehues, A.; Specht, M.; Hippler, M.; Fufezan, C. pymzML - Python module for high throughput bioinformatics on mass spectrometry data. *Bioinformatics* **2012**, *28*, 1052.

(24) Thain, D.; Bent, J.; Arpaci-Dusseau, A.; Arpaci-Dusseau, R.; Livny, M. Pipeline and Batch Sharing in Grid Workloads. *Proceedings of*

the Twelfth {IEEE} Symposium on High Performance Distributed Computing **2003**, 152.

(25) Reisinger, F.; Krishna, R.; Ghali, F.; Ríos, D.; Hermjakob, H.; Antonio Vizcaíno, J.; Jones, A. R. JmzIdentML API: A Java interface to the mzIdentML standard for peptide and protein identification data. *Proteomics* **2012**, *12* (6), 790−794.

(26) Craig, R.; Beavis, R. C. TANDEM: matching proteins with tandem mass spectra. *Bioinformatics* **2004**, *20* (9), 1466−1467.

(27) Bruderer, R.; Bernhardt, O. M.; Gandhi, T.; Miladinović, S. M.; Cheng, L.-Y.; Messner, S.; Ehrenberger, T.; Zanotelli, V.; Butscheid, Y.; Escher, C.; et al. Extending the Limits of Quantitative Proteome Profiling with Data-Independent Acquisition and Application to Acetaminophen-Treated Three-Dimensional Liver Microtissues. *Mol. Cell. Proteomics* **2015**, *14* (19), 1400−1410.

(28) Kessner, D.; Chambers, M.; Burke, R.; Agus, D.; Mallick, P. ProteoWizard: open source software for rapid proteomics tools development. *Bioinformatics* **2008**, *24* (21), 2534−2536.

(29) Barth, J.; Bergner, S. V.; Jaeger, D.; Niehues, A.; Schulze, S.; Scholz, M.; Fufezan, C. The interplay of light and oxygen in the reactive oxygen stress response of Chlamydomonas reinhardtii dissected by quantitative mass spectrometry. *Mol. Cell. Proteomics* **2014**, *13* (4), 969−989.

(30) Wiśniewski, J. R.; Zougman, A.; Nagaraj, N.; Mann, M. Universal sample preparation method for proteome analysis. *Nat. Methods* **2009**, *6* (5), 359−362.

(31) Liu, N. Q.; Dekker, L. J. M.; Stingl, C.; Güzel, C.; De Marchi, T.; Martens, J. W. M.; Foekens, J. a.; Luider, T. M.; Umar, A. Quantitative proteomic analysis of microdissected breast cancer tissues: Comparison of label-free and SILAC-based quantification with shotgun, directed, and targeted MS approaches. *J. Proteome res.* **2013**, *12* (10), 4627−4641.

(32) Yates, J. R.; Park, S. K. R.; Delahunty, C. M.; Xu, T.; Savas, J. N.; Cociorva, D.; Carvalho, P. C. Toward objective evaluation of proteomic algorithms. *Nat. Methods* **2012**, *9* (5), 455−456.

(33) Gibbons, B. C.; Chambers, M. C.; Monroe, M. E.; Tabb, D. L.; Payne, S. H. Correcting systematic bias and instrument measurement drift with mzRefinery: Fig. 1. *Bioinformatics* **2015**, btv437.

(34) Cox, J.; Mann, M. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat. Biotechnol.* **2008**, *26* (12), 1367−1372.

(35) Ma, K.; Vitek, O.; Nesvizhskii, A. I. A statistical model-building perspective to identification of MS/MS spectra with PeptideProphet. *BMC Bioinf.* **2012**, *13 Suppl 1* (Suppl 16), S1.

(36) Bruderer, R.; Bernhardt, O. M.; Gandhi, T.; Miladinović, S. M.; Cheng, L.-Y.; Messner, S.; Ehrenberger, T.; Zanotelli, V.; Butscheid, Y.; Escher, C.; et al. Extending the Limits of Quantitative Proteome Profiling with Data-Independent Acquisition and Application to Acetaminophen-Treated Three-Dimensional Liver Microtissues. *Mol. Cell. Proteomics* **2015**, *14* (5), 1400−1410.