**RUHR-UNIVERSITÄT** BOCHUM

**Ruhr University Bochum**
**Medical Faculty**
Medical Proteome Center
Medical Bioinformatics / Functional Proteomics
**Center for Protein Diagnostics (PRODI)**

Bundesministerium für Bildung und Forschung
FKZ 031L0201

# Enhancements of MaCPepDB – the Mass Centric Peptide Database

Dirk Winkelhardt, Martin Eisenacher, Katrin Marcus, Julian Uszkoreit
Ruhr University Bochum, Medical Faculty, Medical Proteome Center
Ruhr University Bochum, Center for Protein Diagnostics (PRODI), Medical Proteome Analysis

## Abstract

Often challenged with small amounts of samples, researchers who want to run targeted proteomic experiments like single-, multiple- and parallel reaction monitoring (SRM, MRM and PRM) need to know their specific targets before even measuring their samples once. Furthermore, it is important to know, whether selected peptides are unique, at least for the species of interested.
To these lengths we developed MaCPepDB [1] (Mass Centric Peptide Database). MaCPepDB contains the tryptic in silico digest of all known proteins in UniProt KB, stored in an efficient manner to be quickly searched.
For the upcoming release of MaCPepDB we were able to increase the performance and stabilize the response times with the help of modern distributed database technologies, as a result the number of concurrent users is improved. This performance gain allowed us to provide additional data for each peptide. One of these improvements is a list of taxonomies for each peptide, which highlights whether the peptide is unique or shared in each respective species.
We are also developing additional tool based on MaCPepDB. An example is a search engine, called MaxDecoy, which is able to perform spectrum identifications against the complete UniProt KB in roughly 9 hours. Very preliminary results are a shown below.

## Performance improvements by Citus Data

The original MaCPepDB was build on a single server with 112 Cores, 754 GB RAM and 3 consumer SSDs in a RAID 5. This setup quickly reaches the limits of its I/O at roughly 540 MB/s when published.
The second generation of MaCPepDB was built using Citus Data, extending PostgreSQL, to distribute and utilize multiple servers. The new implementation was tested first on our inhouse OpenStack with 6 virtual machines, each with 32 core and 128 GB RAM. The data was stored on SAN-storage with 32 SSDs in RAID 6 connected using iSCSI over four parallel 10 GBit network, in theory allowing a theoretical throughput of 5 GB/s.
A second test was performed on a database cluster assembled of five 10 year old Dell servers, each with 24 to 32 cores and 128 GB RAM using two consumer SSDs in RAID 0 as main storage for the database and a sixth more recent server which was used as controller to store a few GB of metadata for the database and distribute the incoming queries. In total the second cluster matches the throughput of the first one.
Each cluster performed an in-silico digest of the complete UniProt KB to test the write performance while the read performance was tested by querying 26,897 MS2 precursors from one of our standard measurements including the two most common post translational modification (static carbamidomethylation of Cysteine and variable oxidation of Methionine) which results in a total of 981,566 masses to query.
As Figure 1 shows, the OpenStack-solution needed 58.92 days to build the database with an average of 42.19 inserted proteins per second. Surprisingly the old Dell server cluster was much faster: it needed only 37.15 days with an average of 70.27 proteins per second to complete the database build.

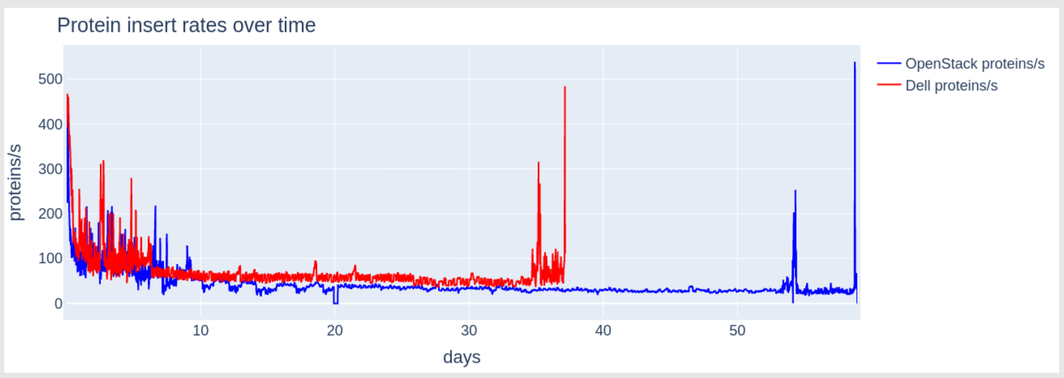


**Fig. 1:** Inserted proteins over time when building the database

The Dell servers do not only perform better in writing data, but also in reading. While the Dell servers queried the 26,897 precursors in roughly 9 hours, the OpenStack-solution was aborted after 65 hours as shown in Figure 2.
Despite the same throughput, there is a huge difference between both clusters which could be explained by the used iSCSI volumes paired with a large MTU of 9000 bytes resulting in mostly empty and delayed network packages when reading small amounts of data.

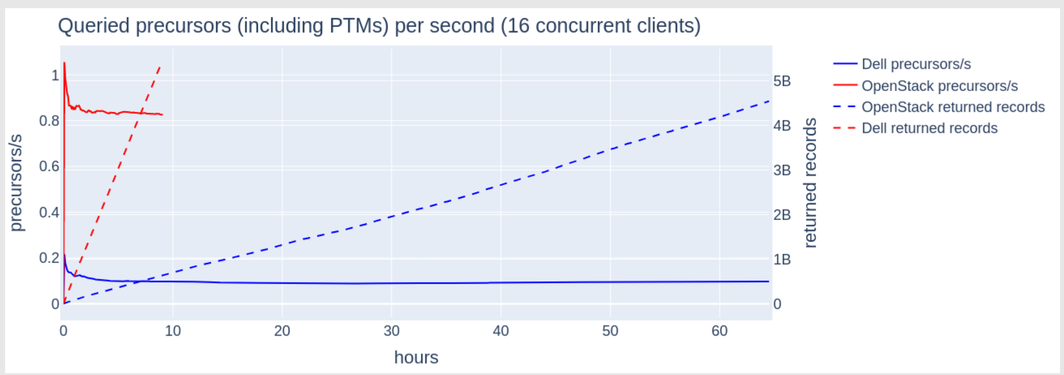


**Fig. 2:** Queried precursors (including PTMs) per second and returned records

## Additional data

With the ability to extend the database even further, by simply adding more servers to a cluster, it becomes possible to add additional peptide information to MaCPepDB.

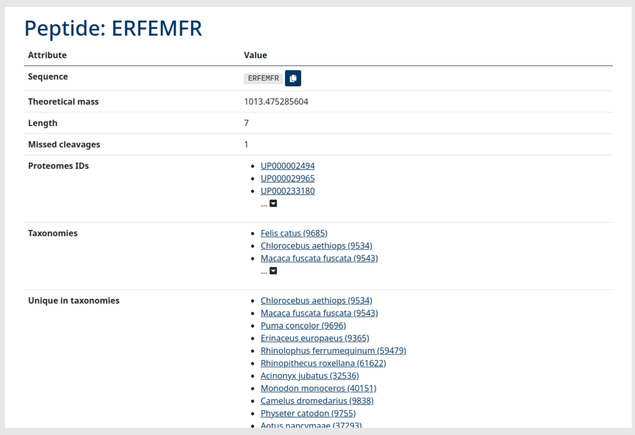


**Fig. 3:** Peptide “ERFEMFR” with new proteome and taxonomy lists

For now lists of taxonomy and proteome IDs are added to each peptide entry which show the respective peptide’s species and proteome information. This also includes the list “Unique in taxonomies” (Figure 3), which highlights the taxonomies where a respective peptide is only contained in one protein, which e.g. makes it a candidate for targeted proteomics approaches.

## MaxDecoy: Improved spectrum identification with MaCPepDB

After the actual search engine’s peptide identification, often a strategy containing the target-decoy-approach to estimate the false discovery rate (FDR) is applied. While this strategy worked well for many years, new high-resolution mass spectrometers with precursor and fragment mass errors in the lower ppm respective mmu range exhibit problems. Firstly, the essential decoys are no longer identified, as their theoretical mass spectra do not fit the measured data. With this effect, the traditional FDR estimation is no longer possible. Furthermore, almost all search engines perform well in distinguishing which given peptide matches a spectrum best. But the differentiation, whether the match of one spectrum is better than another spectrum’s match, is often not possible when sing the algorithm’s scores. Many search engines have for example a tendency to score heavier, longer peptides higher than lighter, smaller sequences. Additionally, for analyses using very large search spaces, like metaproteomics or open searches, the FDR overestimation yields less identifications than default approaches.
To overcome these problems, we modified and applied a compute-intensive strategy introduced in 2015 [2], which can now be applied using cloud technology approaches. Instead of matching only the relatively few peptides in the precursor tolerance to each respective spectrum, we additionally match thousands of decoy peptides per spectrum, which are specifically created to match the spectrum’s tolerance. This amount of peptide spectrum matches per spectrum will allow us to calculate well-calibrated e-values per spectrum, which are comparable between spectra and hopefully require no additional FDR estimation. As a side-effect of our strategy we can allow searches with very large databases – up to the complete UniProt KB – without exhibiting the FDR problems, which currently lead to lower sensitivity.
While we are working on a way to calculate a score or p-value like metric for the new spectrum identification, the basics of the new approach is implemented using Comet as the actual search engine.
Further improvements of this approach will include the usage of predicted MS2 spectra for the scoring of peptide spectrum matches.

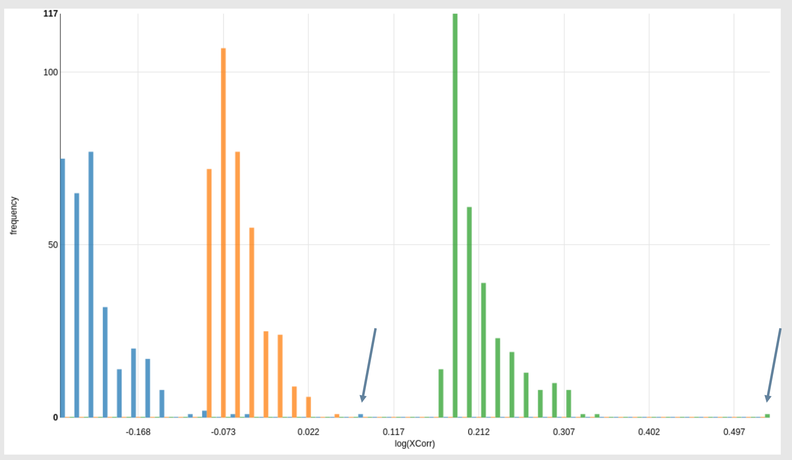


**Fig. 4:** Distribution of log(XScore) for the identifications of three spectra. Blue and green have both one outlier, which shows one good identification for the spectra, while blue generally yielded into much lower scores for all possible IDs than green. Orange has mediocre scores at all and no designated outlier.

References:
1. MaCPepDB: https://doi.org/10.1021/acs.jproteome.0c00967
2. On the importance of well-calibrated scores for identifying shotgun proteomics spectra. Keich U, Noble WS. J Proteome Res. 2015 Feb 6;14(2):1147-60. doi: 10.1021/pr5010983. Epub 2014 Dec 17.