An end-to-end machine learning workflow for ms-based proteomics

Siegfried Gessulat, Tobias Schmidt, Michael Graber, Samia Ben Fredj, Lizi Mamisashvili, Patroklos Samaras, Florian Seefried, Magnus Rathke-Kuhnert, Daniel P. Zolg, Martin Frejno
MSAID GmbH, Garching b. München, Germany

MSAID
Transforming Proteomics

+ Overview

Machine learning augments various steps in proteomics data analysis, from training models for peptide properties and predicting them, to the usage of target/decoy classifiers as in Percolator[1] for error control. Generating, evaluating, and integrating such models largely remains unautomated manual work. Here we present an end-to-end workflow that automates the steps from raw data to production-ready model.

+ Workflow

1 Data ingest

Setting-up an ingest requires meta information that is not always available on PRIDE.

- Download public spectrum and search files from PRIDE.
- Indexing: infer relationship of spectrum and search files
- Ingest: convert to standard formats and upload to AWS S3

2 Data preprocessing

Preprocessing is purpose- and model-specific.

- Filtering: focus on high-quality data, outlier removal
- Data harmonization: aligned to standards and normalization
- Dataset preparation: format conversion, shuffle, splitting

3 Model training

Model definitions and their evaluation are purpose-specific. Architecture search and hyperparameter search are not.

- Architecture: layers and depth of the models is sampled
- Optimization: hyperparameters are sampled
- Training: architecture-hyperparameter combinations are trained
- Evaluation: purpose-specific scripts evaluate each model

4 Model export

- Optimization for client or server-side
- Platform-specific export

1. Data ingest

Spectra, search results and their relations are ingested to a data lake (AWS S3). Sequences are converted to the Proforma standard. Information is accessible interactively via Jupyter Notebooks or RStudio. All ingests are logged (see below).

Example: ProteomeTools[2] (synthetic peptides dataset)

+ Original size: ~9TB
+ Total ingest time: 23h
+ ~110M unique PSMs
+ Compressed size: ~890GB
+ AWS S3 cost: ~20$ / month
+ ~9M precursors

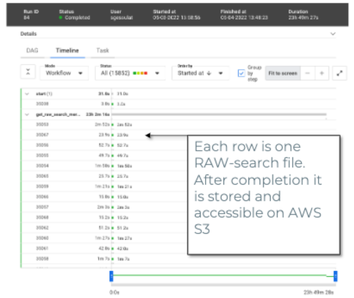


Figure 1: log of ProteomeTools search and spectra ingest to the data lake.

+ 15,852 RAW-search files
+ 1,5 min processing time per RAW-search file.
+ Parallelization in our cluster with 16 workers brings down processing time to <24h
+ > 700 RAW files per hour

2. Data preprocessing

Example: fragmentation spectrum prediction

For fragmentation we annotate y-, b-, immonium, parent and the most frequent combinations of neutral loss ions with charges 1-3. Data is written to tfrecords files. The data is split in train, test, and validation while preventing duplication of peptides across splits.

Figure 2: log of the fragmentation data preprocessing.
Running on a server with 32 workers takes ~5h for the 15,852 RAW-search files of ProteomeTools

3. Model training

The model architecture and hyperparameter search generated, trained, and evaluated >2,500 distinct models within the last year. Trainings are logged along with the respective model evaluation.

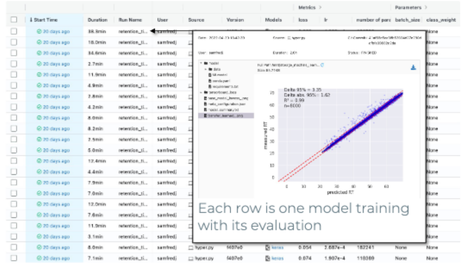


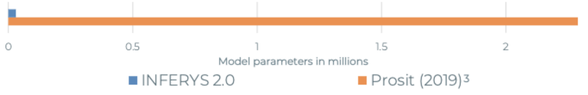
Figure 3: log of retention time models trained with architecture search and hyper-parameter optimization

Retention time prediction

The retention time model is calibrated to a specific dataset via refinement learning. The workflow automatically trains a base model and refines it on 4 external test datasets. Then, the refined model is evaluated on those datasets and compared to baseline models (Figure 3). We identified a model that is substantially smaller than other state-of-the-art models but similarly accurate.

+ Time for refinement learning on a 60-min run: <1 min (CPU)
+ Prediction time for a human digest: <2 min (GPU)

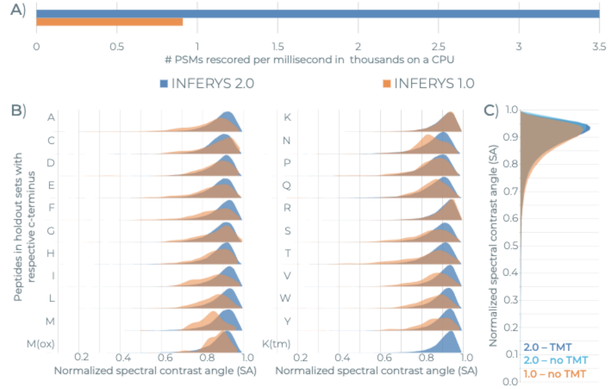
Figure 4: retention time model size comparison



Fragmentation spectrum prediction

This year we integrated TMT and CID spectra and improved prediction accuracy for HLA peptides. At the same time the model size was reduced by a factor of 4 resulting in a 3-fold speed-up.

Figure 5: fragmentation model evaluation. **A**) speed **B**) c-termini **C**) TMT



4. Model export

Models are exported for usage within client software (via C++ or Python-bindings) or remotely via tensorflow serving (gRPC). Models can be encrypted and are optimized for NVIDIA GPUs, Intel, AMD, or ARM CPUs.

+ Integrations

Fragmentation and retention time models generated by this workflow are integrated into the intelligent search algorithm **CHIMERYS™**, a software node in Thermo Scientific™ Proteome Discoverer™ (PD). A fragmentation model is integrated in **INFERYS™** spectral library generation (in PD) and **INFERYS Rescoring** (PD Node).

+ Related Talks

Monday Oral (today): "A streamlined rescoring implementation for comprehensive proteomic data processing" by Daniel Zolg at 2:50pm in the Informatics: Peptide and Protein ID session.

Thursday Oral: "A unifying, spectrum-centric approach for the analysis of peptide tandem mass spectra" by Martin Frejno at 8:50am in the Informatics: DIA and Multiplexing session.

MSAID BOOTH #526 | www.msaid.de/asms-2022

References [1](The et al 2016) "Fast and Accurate Protein False Discovery Rates on Large-Scale Proteomics Data Sets with Percolator 3.0" [2](Zolg, Wilhelm et al 2017) "Building ProteomeTools based on a complete synthetic human proteome" [3](Gessulat, Schmidt et al 2019) "Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning"