Use of carry-over models in false-discovery rate filtering in real-time search

Jesse Canterbury, William Barshop, Graeme McAlister, and Philip Remes, Thermo Fisher Scientific, San Jose, CA 95134

ABSTRACT

Purpose: Here we explore the value of using a false discovery rate estimator model, trained in real-time on one LC-MS run, and either used in subsequent runs as a static model – i.e. *carried over* to the next run -- or else allowed to train further.

Methods: MS2 spectra are searched in real time and results filtered according to a linear-discriminant score chosen at a given FDR, estimated by matches to forward and reverse peptide sequences. The specific weights for the LDA determined in an initial run are allowed to be used in subsequent runs. *It should be emphasized that this approach comprises a data acquisition strategy and is not meant to replace or match offline analysis.*

Results: While results did not yield substantial increases in terms of protein quantifications, we show changes in other metrics matching expectations. Overall, these results show potential for application in different sample contexts, as well as validating the model approach.

INTRODUCTION

Two years ago we introduced the first commercially available platform employing real-time database search with results filtered by an online false discovery rate (FDR)-based estimator [1]. The filter, employing linear discriminant analysis (LDA), must be trained in the initial part of an LCMS run, thus potentially sacrificing some early-run sensitivity in order to maximize sensitivity in later parts of the run. In this work, we describe carrying the filter's feature weights over from run-to-run, using the output weights of one run as input weights in a subsequent run, effectively skipping the initial training step for those later experiments.

MATERIALS AND METHODS

LC-MS. Using a Thermo Scientific™ EASY-nLC™ 1200, we ran LC gradients of about 2 hours in length, ramping buffer B (95:4.9:0.1 acetonitrile:water:formic acid) from 5% to 35%. 250 ng of peptides originating from the Pierce TMT11plex Yeast Digest Standard were loaded onto a 30 cm fused silica column, 75μm inner diameter, packed with C4 reverse phase material. A homebuilt nanoLC source was coupled to a modified Thermo Scientific™ Orbitrap Eclipse™ mass spectrometer.

Acquisition software. We modified the standard Tribrid Series instrument control software (ICSW), which has featured real-0time search capabilities for several years. For real-time interpretation of MS/MS spectra, the ICSW uses a version of the Comet search engine [3]. Spectra are packaged by the mass spectrometer's embedded software and sent to the host PC for processing by Comet, which is contained in a Windows service controlling all aspects of the PC-side real-time processing. LDA training and score determination are performed within the same service. Normally, the LDA is trained in real-time for each run individually. In this case, our modifications allowed LDA parameters (weights) to be carried over into the next run.

After processing, search results are then packaged and returned to the embedded software, where results are filtered according to score criteria, and matching peaks are selected for subsequent synchronous precursor selection (SPS) and MS3 fragmentation.

Data Analysis. For offline analysis, RAW files were analyzed with Thermo Scientific™ Proteome Discoverer™ (PD) 2.5. Further custom analyses were carried out using code written in Lua or C#.

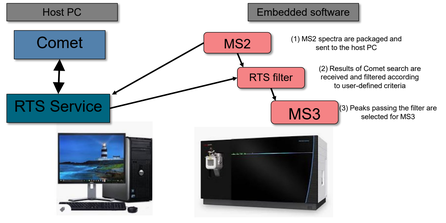


Figure 1. Overview of the real-time search architecture.

Real-time spectral scoring based on FDR

PSM evaluation based on linear discriminant score

Following the approach described by Schweppe et al. [2], we reduce a set of scores and metrics to a single score, using linear discriminant analysis (LDA). For computing the LDA we used the Accord.NET Framework, an extremely versatile library for machine learning [4]. Our feature set contains results of the Comet search as well as other separate features: Xcorr, dCn, precursor mass accuracy, fraction of ions matched, fraction of ion current explained, peptide length, and charge state.

Figure 2 (below). During an LC-MS run we accumulate target and decoy PSMs up to a specified minimum, compute the LDA, score all PSMs according to feature weights determined by the LDA, and then determine the score threshold needed to maintain a 20% false discovery rate. This process is repeated every 1000 target PSMs (cumulative), and the score threshold is updated. After the initial discriminant calculation, new PSMs are no longer filtered according to hard cutoffs of Xcorr, dCn, etc., but instead are filtered according to the LDA-derived score.

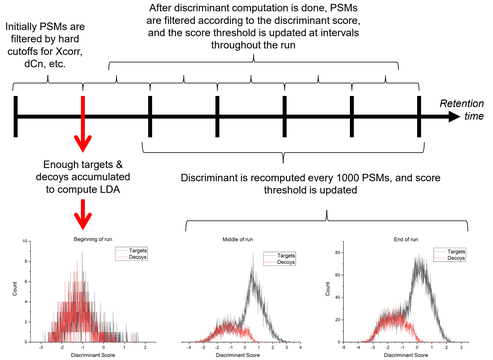


Figure 3. At right we show the time evolution of the LDA feature weights throughout the LC-MS run. In this case, the ion percent weight was most significant, followed by Xcorr, and dCn, the latter of which assumed an increasingly important role as the run progressed.

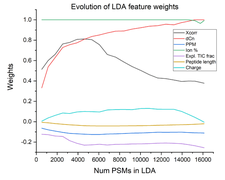
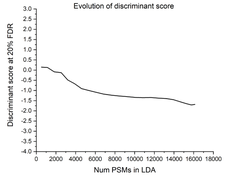


Figure 4. At right we show the evolution of the discriminant score itself, a linear combination of the above features. While the feature weights show some significant changes, the score itself does not actually vary appreciably during the run, an observation that is important for our conclusions in this work.



Static model carry-over

LDA weights are frozen at the end of a first run, and carried over to the next

In a *static* model carry-over, the feature weights are taken from the end of the previous run, here termed the **base** run and applied without alteration to the following run, here termed the **carry-over** run.

Figure 5. At right we show the rate of acquisition of MS3 scans, which are gated by a positive result from the real-time search filter. There are gains to be seen relative to the previous run in the first 60 mins, where the weights were being trained in the base run.

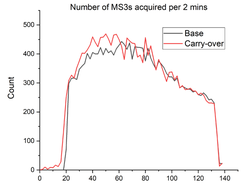


Figure 6. At right we show the number of PSMs passing the discriminant score set for 20% false discovery rate. There is a slight gain in the carry-over run.

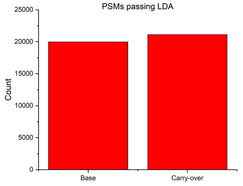


Figure 7. At right we show the number of quantified peptides (peptide groups with TMT abundances > 0 in any channel), from an offline analysis done using Proteome Discoverer (Sequest HT and Percolator).

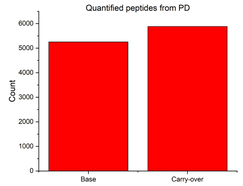
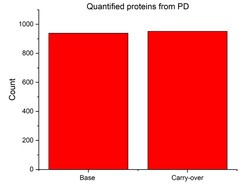


Figure 8. At right we show the number of quantified proteins (protein groups with TMT abundances > 0 in any channel), from an offline analysis done using Proteome Discoverer (Sequest HT and Percolator).



Model continuation

LDA training continues from run to run

In a *continuation* model carry-over, the feature weights are refined throughout all runs, from the **base** run through subsequent **continuation** runs.

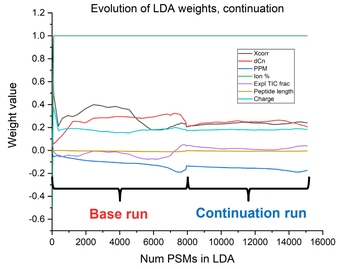


Figure 9 (above). Above we show the evolution of the LDA weights between 2 adjacent runs, where the weights are continuously refined throughout both runs. There is a clear discontinuity when the new run starts.

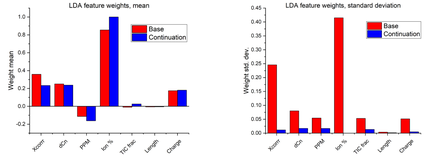


Figure 10 (above). Above we show the mean & standard deviation of each feature weight, comparing the base vs. the continuation run. The mean values show some small changes, whereas the standard deviation is greatly reduced in the continuation run.

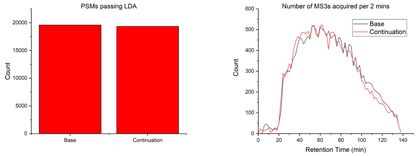


Figure 11 (above). Above we show some metrics from the base & continuation runs. The number of PSMs passing the discriminant score actually decreases slightly from the base to the continuation run. Looking at the rate of MS3 acquisition in the continuation run, we see that there were slightly more MS3s acquired in the earlier part of the run compared to the base run, and slightly fewer in the last 40 minutes.

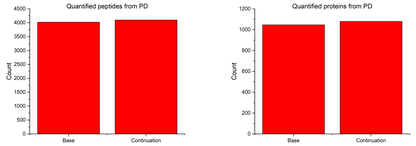


Figure 12 (above). Above we show the slight gains in quantified peptides and proteins in the continuation run relative to the base run. Unfortunately, these numbers are likely well within the run-to-run variance.

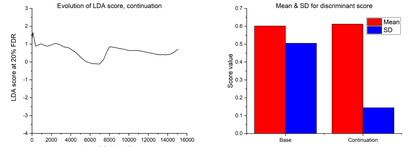


Figure 13 (above). The evolution of the LDA score shows little variation between the base and continuation runs, which likely explains the modest gains shown above.

CONCLUSIONS

- Use of carry models can be a useful way to circumvent the training stage in online PSM filtering by FDR estimation. The gains shown here are modest at best, reflecting the robustness of the LDA model employed and the speed with which it converges on a suitable solution.
- Users of the "Enable FDR Filtering" feature in Real-Time Search need not be concerned that training the LDA will sacrifice results.
- It is expected that this approach will be more useful in situations where the LDA weights do not converge so quickly, for example in the context of samples that are expected to yield smaller numbers of high-scoring PSMs than those shown here.

REFERENCES

1. Canterbury et al., Poster MP112, ASMS 2020.
2. Schweppe et al., *J. Proteome Res.* 2020, **19**, 2026-2034; Erickson et al., *J. Proteome Res.* 2019, **18**, 1299-1306.
3. Eng et al., *JASMS* 2015, **26**, 1865..
4. Accord.NET Framework v. 3.8.0, available at http://accord-framework.net.

ACKNOWLEDGEMENTS

We gratefully acknowledge Devin Schweppe and Chris McGann of the University of Washington for helpful and insightful discussions on real-time search and online FDR estimation. We also gratefully acknowledge the assistance of members of Mike MacCoss's laboratory at UW for their assistance with instrumentation and further discussions, especially Lilian Heil and Rich Johnson.

TRADEMARKS/LICENSING