Semi-supervised learning for nucleic acid cross-linking mass spectrometry

Emil Paulitz

July 24, 2020

Contents

1 Introduction

- Motivation

2 Background

- Proteomics, MS, Peptide Identification, Protein-RNA cross-linking (alles relativ kurz)

Algorithms like Sequest [3] or X! Tandem [2] compare the resulting spectra with theoretical spectra calculated from every possible peptide generated from

a given protein list and compute a score based on their similarity. The best scoring peptide is then considered a peptide-spectrum-match (PSM). The score produced by those algorithms often do not distinguish well enough between correct and incorrect matches [5], but their scores enable FDR estimation using decoy databases and serve as a basis for score re-calibration with the Percolator algorithm [4].
Decoy databases are created from the target database, contain usually as many peptides [7, 6] in a reversed or shuffled order with respect to the amino acid sequence [1]. They are presented to the scoring algorithm either separately [4] or mixed with the target database [7]. It is assumed, that decoy and target peptides have similar features [6] and are not easily distinguishable by a scoring algorithm. When the actually fitting peptide for a given spectrum is not in the target database, and thus a wrong one will be chosen, the best scoring peptide will be a decoy approximately half of the time. This allows for an estimation of wrongly assigned targets, since the score distribution is assumed to be the same for decoys and false targets [1].
In practice, one estimates the probability of a PSM being a false target by counting the number of decoy-PSMs with the same or a higher score. It is then assumed, there are as many false targets and thus a false discovery rate (FDR) can be estimated. This leads to the following formula [4]:

$$FDR = \frac{\#\text{ false target PSMs}}{\#\text{ all target PSMs}} \approx \frac{\#\text{ decoy PSMs}}{\#\text{ all target PSMs}} \tag{1}$$

In this thesis, the following approximation is used:

$$FDR \approx \frac{\#\text{ decoy PSMs}}{\#\text{ all PSMs}} = \frac{\#\text{ decoy PSMs}}{\#\text{ decoy PSMs} + \#\text{ target PSMs}} \tag{2}$$

It is faster to calculate and yields results differing by the FDR, so in the relevant range of FDRs of 0 to 5% up to 5%:

$$\frac{\frac{\#\text{ decoys}}{\#\text{ targets}}}{\frac{\#\text{ decoys}}{\#\text{ decoys}+\#\text{ targets}}} = \frac{\#\text{ decoys} + \#\text{ targets}}{\#\text{ targets}} = 1 + \frac{\#\text{ decoys}}{\#\text{ targets}} \approx 1 + FDR \tag{3}$$

The q-value as a measure for a single PSM rather than a metric for a set of PSMs is then derived from this as the minimum FDR of all PSMs with a lower or equal score [4, 1]. It will be used for estimating the credibility for any one PSM.
As Käll et al. [5] say, separating correct from incorrect target PSMs with already mentioned algorithms works fine, but there is still room for improvement. This is because often not all information is used and considered jointly. Percolator [5, 4] tries to utilize as much information as possible by using scores from different algorithms, features of the peptide like its length, of the spectrum or the PSM itself. It joins them using a linear SVM and a semi-supervised approach with cross-validation to retain as many PSMs as possible. In every iteration, the top ranking, non-decoy PSMs up to a certain threshold of q-value are chosen as positive training examples, and the decoy PSMs are used as negative training set. The PSMs are then re-ranked using the SVM score, with the intend of getting a better separation of true and false PSMs. If that holds true, the positive training set of the next iteration better is of higher quality and the

SVM can be trained even better. The algorithm usually converges within the first 10 iterations [5]. To avoid having to split the set into training and testing set and consequently losing possibly correct PSMs but also avoid overfitting, currently a nested cross-validation approach is being used [4].

3 Material, Methods

- Material: Was ich für ein Datensatz zum Testen benutzt habe und wo der herkommt

3.1 Implementation of the percolator algorithm

- Wie genau stelle ich das vor, siehe Mail? Wichtige Punkte wären:
- Verwendete Scipy-Methoden
- Abbruch wenn es nicht besser wird und dass ich die AUC als Metrik nutze
- feature normalization
- Wichtige Hilfsfunktionen (pseudoROC zB)

3.2 Improvements of the percolator algorithm for cross-link identification

- Klassenspezifische q-values
- ROC nach jeder Iteration und aufgesplittet nach XL/nXL

3.2.1 How to deal with different Ranks

- OptimalRanking Option (Erst paar Iterationen Ränge verändern lassen und dann die schlechten entfernen)

3.2.2 Characteristics of cross-linking PSM datasets

- Verhältnis Targets:Decoys und XL:non-XL in inneren und äußeren splits gleich lassen und MinMaxMedian Auswertungen mithilfe von google colab cloud computing
- Imputation (kam zwar nichts raus ist aber trotzdem interessant)
- Trennung von Datensatz nach XL/nXL oder sogar cross-linking target falls Datensatz groß genug

3.2.3 Small datasets

- Ratio Testing (nicht-random aus ganzem Datensatz und random aus Top 10%. Liefert Erkenntnisse über die mögliche Größe des Datensatzes und eventuell die Sinnhaftigkeit, wann man die Datensätze einfach trennen kann → Für den Leser relevant)
- Einbau von Identifikationen bei 1% FDR als Metrik (Sinnhaftigkeit kann man ja diskutieren)

(- Performance auf anderem Datensatz
- Vergleich mit Entrapment FDR)

4 Results

4.1 Implementation of the percolator algorithm

- Reimplementierung funktioniert wie Original
- feature normalization war wichtiger boost
- ROC nach jeder Iteration zeigen

4.2 Improvements of the percolator algorithm for cross-link identification

4.2.1 How to deal with different Ranks

- Ergebnisse von OptimalRanking

4.2.2 Characteristics of cross-linking PSM datasets

- Verhältnis Targets:Decoys und XL:non-XL verringt die Streuung: MinMax-Median Auswertungen
- Bei Imputation kam nichts heraus
- Großer Unterschied wenn man den (großen) Datensatz nach XL/nXL oder sogar cross-linking target aufteilt

4.2.3 Small datasets

- Sinnvolle Plots zu Ratio Testing
- Neue Metrik erlaubt es der Implementierung, auch auf kleineren Datensätzen zu funktionieren

5 Discussion

- Methoden hinterfragen oder begründen, Ergebnisse interpretieren, Anwendbarkeit diskutieren, z.B.:
- Warum habe ich mich mit Rängen beschäftigt? (Bei XL-Datensätzen oft sinnvoll um erst percolator entscheiden zu lassen was auf Rang 1 steht (?)) - C Parameter für jeden split neu optimieren führt zu overfitting? → Original-Algorithmus macht es auch so
- Wie sinnvoll ist die neue Metrik? - ScanNr Versuche: Gleiche Spektren (identifiziert anhand der ScanNr.) auf verschiedene splits verteilen verändert nichts, d.h. vermutlich sind die niedrigeren Ränge dann so schlecht, dass es nichts bringt die schonmal gesehen zu haben.
- Peptide Versuche: Schlechtere Ergebnisse, aber vllt ehrlicher?
- Mögliche weiterführende Experimente: mächtigere Klassifikatoren + monotonic constraints (wie von Timo ausprobiert), Ada-Boosting, feature selection

References

[1] Suruchi Aggarwal and Amit Kumar Yadav. False discovery rate estimation in proteomics. In *Methods in Molecular Biology*, pages 119–128. Springer New York, 2016. doi: 10.1007/978-1-4939-3106-4_7. URL https://doi.org/10.1007/978-1-4939-3106-4_7.

[2] R. Craig and R. C. Beavis. TANDEM: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, February 2004. doi: 10.1093/bioinformatics/bth092. URL https://doi.org/10.1093/bioinformatics/bth092.

[3] Jimmy K. Eng, Ashley L. McCormack, and John R. Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, 1994.

[4] Viktor Granholm, William Noble, and Lukas Käll. A cross-validation scheme for machine learning algorithms in shotgun proteomics. *BMC Bioinformatics*, 13(Suppl 16):S3, 2012. doi: 10.1186/1471-2105-13-s16-s3. URL https://doi.org/10.1186/1471-2105-13-s16-s3.

[5] Lukas Käll, Jesse D Canterbury, Jason Weston, William Stafford Noble, and Michael J MacCoss. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods*, 4(11):923–925, October 2007. doi: 10.1038/nmeth1113. URL https://doi.org/10.1038/nmeth1113.

[6] Roger E. Moore, Mary K. Young, and Terry D. Lee. Qscore: An algorithm for evaluating SEQUEST database search results. *Journal of the American Society for Mass Spectrometry*, 13(4):378–386, April 2002. doi: 10.1016/s1044-0305(02)00352-5. URL https://doi.org/10.1016/s1044-0305(02)00352-5.

[7] Junmin Peng, Joshua E. Elias, Carson C. Thoreen, Larry J. Licklider, and Steven P. Gygi. Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (LC/LC-MS/MS) for large-scale protein analysis: the yeast proteome. *Journal of Proteome Research*, 2(1):43–50, February 2003. doi: 10.1021/pr025556v. URL https://doi.org/10.1021/pr025556v.