Comparison of search engines and quantification of the cyanobacterial proteome

Abstract

Analysis of MS data requires efficient processing of big data to provide time and cost savings. Our work was based on the optimization of a workflow in the analytics software OpenMS. The given experimental data was generated in a study about Cyanobacteria and their proteome allocation strategies under light and CO2 limited conditions, made by Michael Jahn et al. We tested the performance of three different search engines MSGF+, Crux and Comet on the data, then used MSGF+ to quantify proteins from Cyanobacteria that had been grown in different conditions.

Introduction

Analyzing MS data is a time consuming process, which requires a lot of computational resources. To analyze raw MS data, it is required to align the fragments with a database, in order to identify which peptides were present in the sample. There are several search engines available for performing the identification, which differs based on different properties. In the first part of this project, we chose to compare three different search engines; MSGFPlusAdapter, CruxAdapter, and CometAdapter. MSGFPlus is an universal search engine for tandem MS data which aims to increase the number of hits by utilizing a "generating function approach". Crux is an database search engine for identification from tandem MS data, that uses an extended version of SEQUEST for database search, and a peptide indexer scheme for speed. Comet is an open source search engine for MS/MS data available on SourceForge. The score function is determined by the scalar product of two vectors representing the input spectrum and calculated fragment ion masses.

The data used in this project is obtained from the article *"Growth of Cyanobacteria is Constrained by the Abundance of Light and Carbon Assimilation Proteins"* by Michael Jahn et.al, published 2018. Cyanobacteria are photosynthetic prokaryotes that play an important role in generation of oxygen on earth and are of great potential for biofuel production due to their carbon-fixation ability. In the article, the authors studied how the proteome of cyanobacteria changes when the cells are exposed to different environmental factors; different light intensity and carbon access. The result of the article give insight in how to optimize the cyanobacteria proliferation, which is of importance in, for example, biofuel production.

What we did

Part 1

In the first part of the project, we decided to use the data provided by Michael Jahn to evaluate three different search engines, MSGFPlus, Crux, and Comet.

Going from raw data to an actual result takes a long time and experience. However, by optimizing the technical workflow and using tools fit for the job, you can build a pipeline for the data which minimizes the amount of time and resources used. The tools themselves also have different properties, which you have to keep in mind when creating a pipeline.

In this project, we used OpenMS for performing the analyzations. OpenMS is an open-source software library for processing LC/MS data and the software provides multiple mass spectrometry related softwares which can be used for analysis, which was perfect for our purpose.

Part 2

In the second part of the project, we used the result from the most efficient pipeline in order to perform a quantification analysis on how the proteome of cyanobacteria changes based on environmental factors; different light intensities or carbon resources. This was performed by identifying proteins related to the photosynthesis in our results, and see how these proteins differed among the samples with different light intensity or carbon access.

The goals

The goal with the project was to analyze the efficiency of different search engines, using raw data from proteomic experiments of cyanobacteria, and use the search engine in a pipeline to perform a quantification analysis.
Although, the main goal with the project for us was to learn how to construct MS analysis workflow, the process of going from raw data to presentable results.

How we did it

KNIME is a software which can be used to quickly create a pipeline from many different modules. By adding OpenMS to KNIME, we could create a workflow for processing the data, using the pipeline in figure 1.
The pipeline took mzML files as output and were then passed to the different search engines. The output was afterwards filtered depending on the FDR values of the PSMs. For MSGFPlus, we also included a step where the PSMs were filtered by *Percolator,* a semi-supervised learning for peptide identification from shotgun proteomics datasets.

Cont.

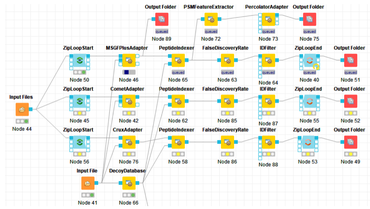

Figure 1: Pipeline for processing the data from mzML to idXML

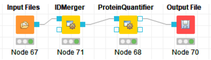

Figure 2: Processing idXML to csv file containing abundances of proteins

The produced files (idXML) were then used for quantification, which is shown in the short pipeline in figure 2. Note that the "ProteinQuantifier" module uses MS2 and not MS1 for quantification, which is another way to quantify proteins. The .csv output file includes a table of all quantified proteins.

To compare the search engines, we summed the amount of peptides mapped to a protein for of each table of quantified proteins, which is shown in figure 2 in our results.

Using the data from MSGFPlus, we processed the data to yield relative abundances of proteins. For each protein, there were two replicates whereas the mean values of the replicates were used for figure 3 to 6. The lowest of the x values (aka 60 µmol photons m^{-2} s^{-1} and 0.15% CO_2) were used to calculate a log2 ratio between each point. To filter out proteins with specific properties and functions, the file containing quantifications of each protein was merged with a file containing annotations for all the proteins, to get the rows containing the proteins we wanted to analyze. The general direction of the log2 ratios were also calculated with linear regression, which is shown as the large black line in each graph.

Our main challenges were due to not having sufficient RAM on our personal computers, so we had both installation and performance problems with different analytics tools. Initially our instruction was to use Quandenser and MaxQuant, but after these programmes crashed several times due to insufficient RAM when attempting to analyse the experimental data, we decided to use KNIME with the OpenMS Plugin. With OpenMS we had to figure out the right settings for every node, so it took a while until we could run a final quantification pipeline.

Our results

Our results showed that MSGFPlus resulted in more peptides matched to proteins, compared to Comet and Crux (see figure 2). MSGFPlus with an additional step that included Percolator, seemed to yield a slightly lower amount of peptides matched. However, this could be due to percolator filtering peptides below FDR < 0.01 in a completely different way compared to usual FDR calculations..

Cont.

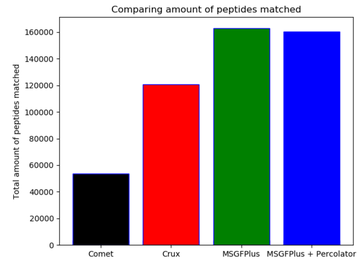

Figure 2: Number of identified peptides with FDR < 0.01

The results from the proteomic part in figure 3 to 6 were as expected, where more sunlight and higher CO_2 levels results in more proteins being synthesized coupled to proliferation and photosynthesis pathway.

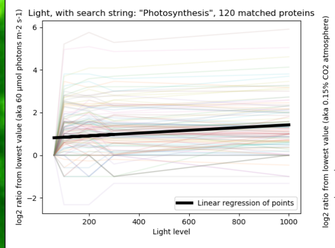

Figure 3: Protein abundance log2 ratio depending on the light level, filtered for the photosynthesis pathway

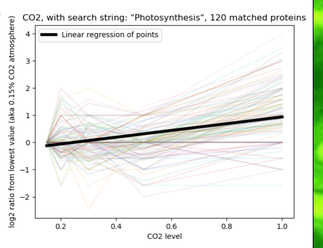

Figure 4: Protein abundance log2 ratio depending on the CO2 level, filtered for the photosynthesis pathway

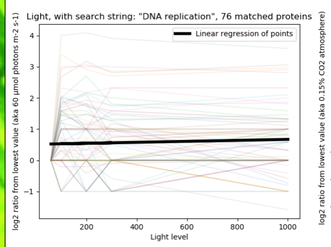

Figure 5: Protein abundance log2 ratio depending on the light level, filtered for the DNA replication pathway

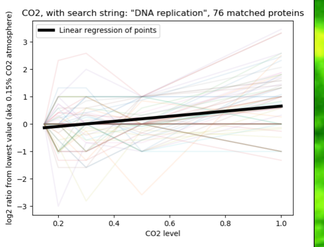

Figure 6: Protein abundance log2 ratio depending on the CO2 level, filtered for the DNA replication pathway

Conclusion

Processing MS data is both time consuming and requires a lot of computational resources. We did not get out a lot of data, but our goal to learn about MS data analyzation was fulfilled, since we now have a better insight of how to process MS data and a functional pipeline which could be used for future use.

In conclusion, the project was a success, even though we had to resort to shortcuts to be able to run the calculations, and the time consuming process of actually trying to get out any results at all.

This project was performed by Timothy Bergström, Stina Borg, and Kristina Schira, with a lot of help from Michael Jahn. Thank you Michael, for making this possible!