OpenSWATH and CCprofiler analysis workflow

System-wide profiling of protein-complexes via size exclusion chromatography– mass spectrometry (SEC-MS) - Methos in Molecular Biology 2020

Fossati A[1] Frommelt F.[1] Uliana F.[1] Martelli C.[1] Vizovisek M.[1]
Gillet L.[1] Collins B.[2] Gstaiger M.[1] Aebersold R[1,3]

Contents

[1] Institute of Molecular Systems Biology, Department of Biology, ETH Zurich, Switzerland
[2] School of Biological Sciences, Queen's University of Belfast, Belfast, UK
[3] Faculty of Science, University of Zurich, Zurich, Switzerland

Introduction

This script provides an overview of commands for the analysis pipeline of protein complexes from SEC-Evosep-SWATH data. For quantitative data extraction the OpenSWATH Workflow (OSW) (Rost et al. 2014) is applied and subsequent statiscal scoring of co-elution proteins with the R-package CCprofiler. To conduct the trial experiment, download the *.mzXML data, the spectral assay library, and iRT file from ProteomeXchange (data set identifier: XXXX). Installation instruction for applied software tools and detailed instructions about commands and parameters are documented on http://openswath.org/en/latest/ .

SWATH data extraction with OpenSWATH

THe OpenSWATH workflow facilitates extraction of quantitative data from SWATH-MS data. For analysis OpenMS v2.4 (downloaded on the 05.09.2019) was used. The extracted peaks are subsequent scored with PyProphet (Rost et al. 2014) and realignment of scored data is performed with TRIC (Rost et al. 2016).

OpenSWATH workflow

```
for file in vmatej_*.mzXML.gz
do
```

```
echo $file TMPOUT=${TMPDIR}/${file%.*.*} mkdir -p ${TMPOUT}
OpenSwathWorkflow -in ${file}
  -tr HS_decoy_EvosepLib.pqp
  -tr_irt combined_iRT_CiRT_201804.TraML
  -Scoring:stop_report_after_feature 5
  -readOptions cache -batchSize 1000
  -min_rsq 0.90 -min_coverage 0.6 -sort_swath_maps
  -batchSize 1000 -extra_rt_extraction_window 60
  -rt_extraction_window 360 -mz_extraction_window 100
  -mz_extraction_window_unit ppm
  -mz_correction_function quadratic_regression_delta_ppm
  -threads 8
  -min_upper_edge_dist 1
  -Scoring:Scores:use_dia_scores true
  -Scoring:TransitionGroupPicker:min_peak_width 10
  -tempDirectory ${TMPOUT}
  -out_osw  ${OUTDIR}/${file%.*.*}.osw
done
```

PyProphet

PyProphet (Rost et al. 2014), is a semi-supervised python based algorithm for scoring OpenSWATH results. For our example we used newest version of PyProphet (version 2.1.4.dev2), which allows the score large-scale dataset (Rosenberger et al. 2017). An extented tutorial is avaialable on the GIT-repository: https://github.com/PyProphet/pyprophet .

To learn the weights for the different scores extracted from OSW, we apply he scoring on ms2 level on the SEC-Input (pre fractionated sample). If several inputs are avaiable, it is advised to generate subsampled merged model.osw file, to learn the scoring weights on a representative sample.

```
pyprophet score
  --in= vmatej_I190208_129.osw
  --level=ms2
```

The learned weights are saved in the input file, which is then used to score all the SEC-fractions. This step can be condcuted in parallel.

```
for run in vmatej_*.osw do pyprophet score
  --in=${run}
  --apply_weights= vmatej_I190208_129.osw
  --level=ms2 done
```

After ms2 based scoring, all fraction files are merged into a single model to score on peptide and protein. The implementation for large-scale datasets allows to subsample from all *.osw files only the necessary columns for experiment-wide scoring.

```
for run in vmatej_*.osw do run_reduced=${run}r
pyprophet reduce
  --in=${run}
  --out=${run_reduced} done
```

To generate a merged file, we have to use as template the spectral library file used to perform quantitation with OpenSWATH. As ouptput we obtain one single model file.

```
pyprophet merge
  --template=HS_decoy_EvosepLib.pqp
  --out=model_global_2.osw *.oswr
```

On the merged file we can perform peptide and protein level error-rate control either per run or global.

```
pyprophet peptide --context=run-specific --in=model_global_2.osw
pyprophet protein --context=global --in=model_global_2.osw
```

The run-specific peptide and global protein error-rates are then transferred back to each individual file.

```
for run in vmatej_*.osw
do
pyprophet backpropagate
  --in=${run}
  --apply_scores=model_global_2.osw done
```

In the last PyProphet step, we can export for each run the OSW results, applying the confidence score thresholds. For SEC data it is beneficial to export the data with higher FDR thresholds, as the data is further filtered more stringently with functions within the downstream analysis pipeline in CCprofiler.

```
for run in vmatej_*.osw
do
pyprophet export
  --in=${run} --no-transition_quantification
  --max_rs_peakgroup_qvalue=1
  --max_global_peptide_qvalue=0.05
  --max_global_protein_qvalue=0.05
done
```

TRIC

Alignment of identified peakgroups across the entire SEC-gradient is perfromed via TRIC (Rost et al. 2016).

```
feature_alignment.py
--in vmatej_*.tsv
--out feature_alignment.csv
--mst:useRTCorrection True
--mst:Stdev_multiplier 3.0
--max_rt_diff 60
--alignment_score 0.05
--target_fdr -1
--max_fdr_quality 0.1
--fdr_cutoff 0.05
--realign_method lowess_cython
--method LocalMST
--disable_isotopic_grouping
```

The resulting feature_alignment.csv is the final OpenSWATH output in long-format and is directely loaded to CCprofiler.

CCprofiler analysis

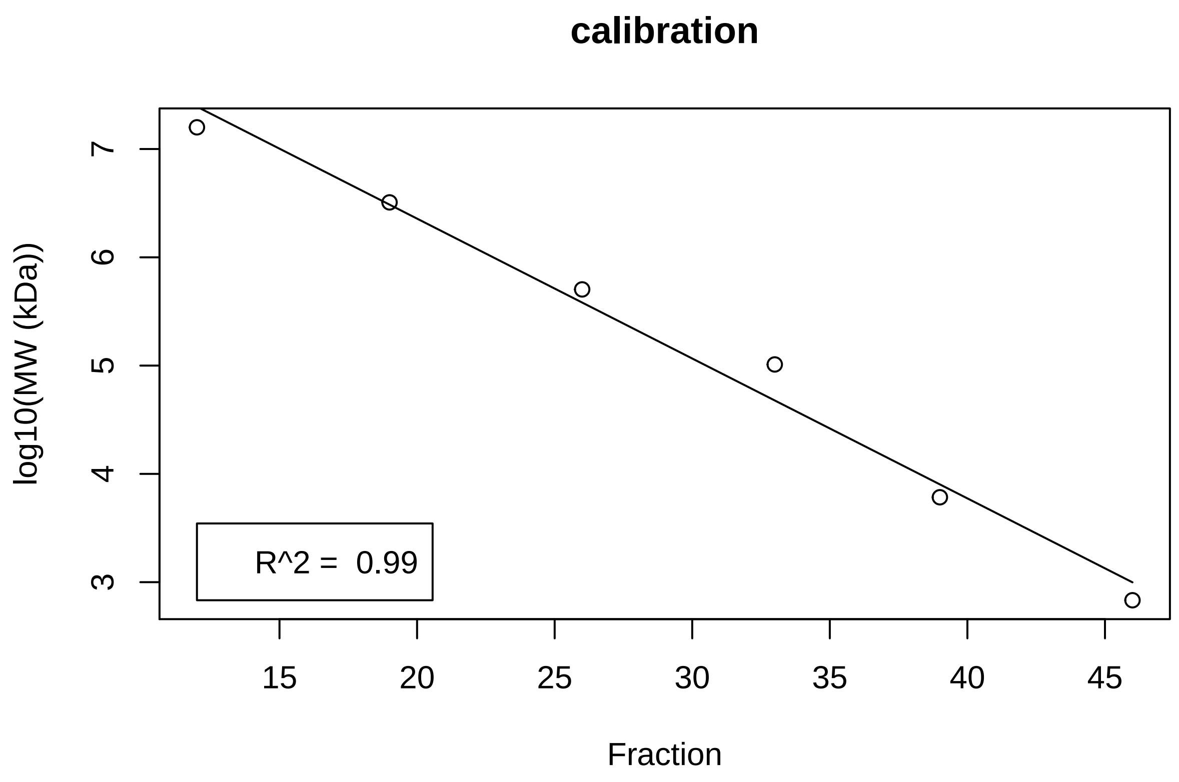
CCprofiler (Heusel et al. 2017) faciliates protein complex analysis from size-based fractionated samples with subsequent MS analysis. For more detailed documentation https://github.com/CCprofiler

R-version used for data analysis CCprofiler git commit d2dc51d

Load required R-packages dependencies for the CCprofiler package.

Data preprocessing before CCprofiler

Different versions of OpenSWATH outputs or datasets analyzed with a different spectral assay library need different preprocessing. The CCprofiler sotware needs only a minium set of identifiers and can be conducted also from DDA data if quantiative data on peptide level is provided. The `'importPCPdata()'` presents a flexible function with minium requirements for loading any PCP-data to CCprofiler, as the function only requires a protein_id, peptide_id, filename, and intensity column. For the presented test data we use the `'importFromOpenSWATH()'` function and first have to add a prefix to identify decoy sequences.



Protein inference in CCprofiler

Finally the file can be imported using the importFromOpenSWATH function. In this step we add the calibration curve as an annotation to the " 'pepTrace' " object.

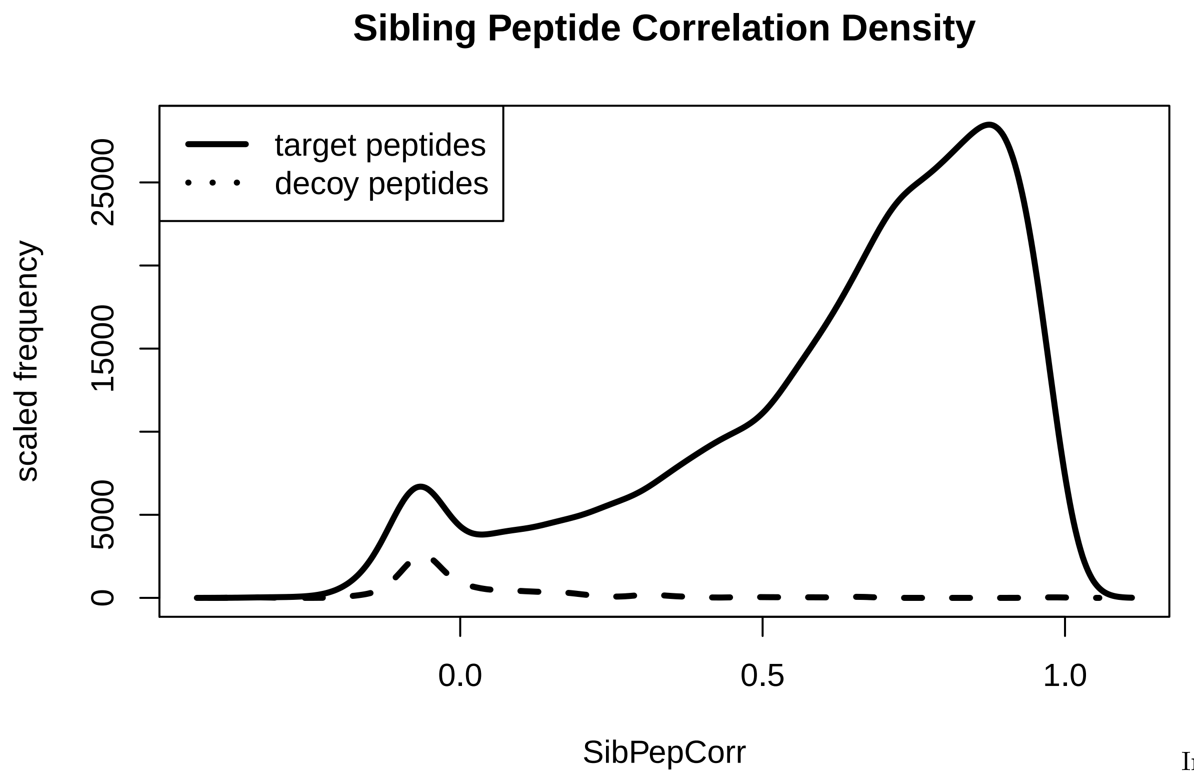
```
pepTraces <- importFromOpenSWATH(data = df,
                                 annotation_table = Annotation_txt,
                                 rm_requantified = TRUE,
                                 MS1Quant = F,
                                 verbose = T,
                                 rm_decoys = F)

pepTraces <- annotateMolecularWeight(pepTraces,
                                     calibration_new)
```

At this step we perform data filtering, sibiling peptide correlation and protein quantitation. First all peptides not identified in minimum of three consecutive fractions are removed. Sibling peptide correlation is a strong filter, as it performs locale correlation for all peptides identified from one protein sequence. Following sibling peptide correlation an absolute correlation cutoff is applied to remove noisy signals.

```
pepTraces_cons <- filterConsecutiveIdStretches(traces = pepTraces,
                                               min_stretch_length = 3)

pepTraces_cons_sib <- filterBySibPepCorr(traces = pepTraces_cons,
                                         fdr_cutoff = NULL,
                                         absolute_spcCutoff = 0.2,
                                         plot = TRUE)
```



In the next step protein quantitative values are inferred from the 2 most abundant peptides peptides.

```
protTraces <- proteinQuantification(pepTraces_cons_sib,
                                    topN = 2,
                                    keep_less = FALSE,
                                    rm_decoys = TRUE)
```

These filtering steps and protein inference results in 1887 unique proteins across the SEC-gradient for which we can now perform protein correlation analysis to obtain protein complexes.

complex centric analysis

```
complexHypotheses <- corumComplexHypotheses
binaryHypotheses <- generateBinaryNetwork(complexHypotheses)
pathLength <- calculatePathlength(binaryHypotheses)
```

```
## Warning in melt(distMatrix): The melt generic in data.table has been passed a matrix and will attemp
## please note that reshape2 is deprecated, and this redirection is now deprecated as well. To continue
## libraries are attached, e.g. melt.list, you can prepend the namespace like reshape2::melt(distMatrix
## error.
```

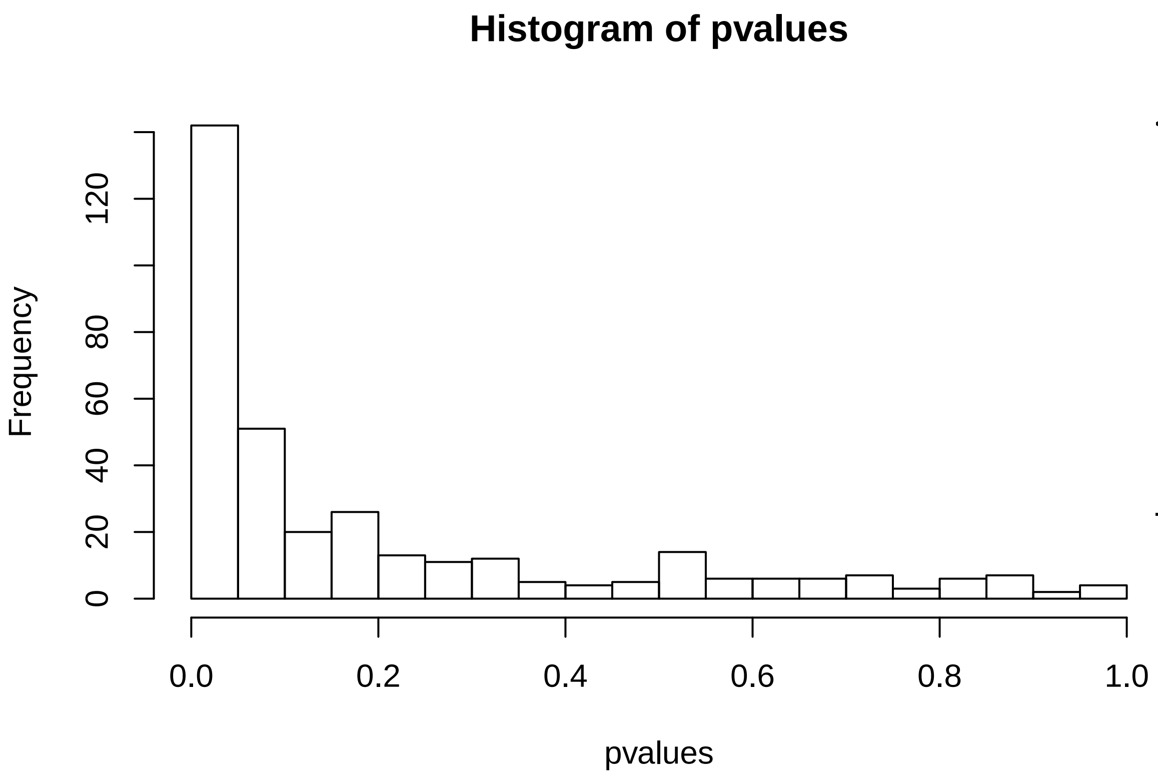
```
corumTargetsPlusDecoys <- generateComplexDecoys(target_hypotheses=complexHypotheses,
                                                dist_info=pathLength,
                                                min_distance = 2,
                                                append=TRUE)
```

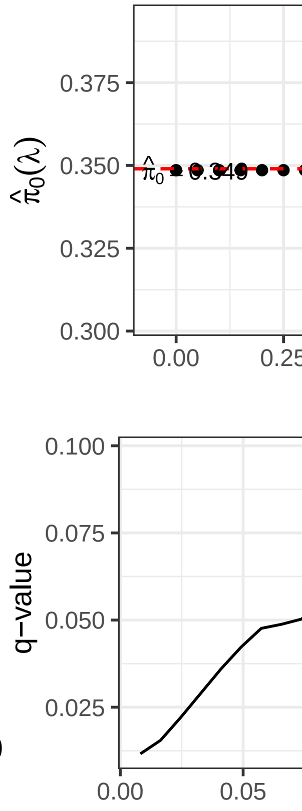
Now we perform coelution score calculation and FDR estimation.

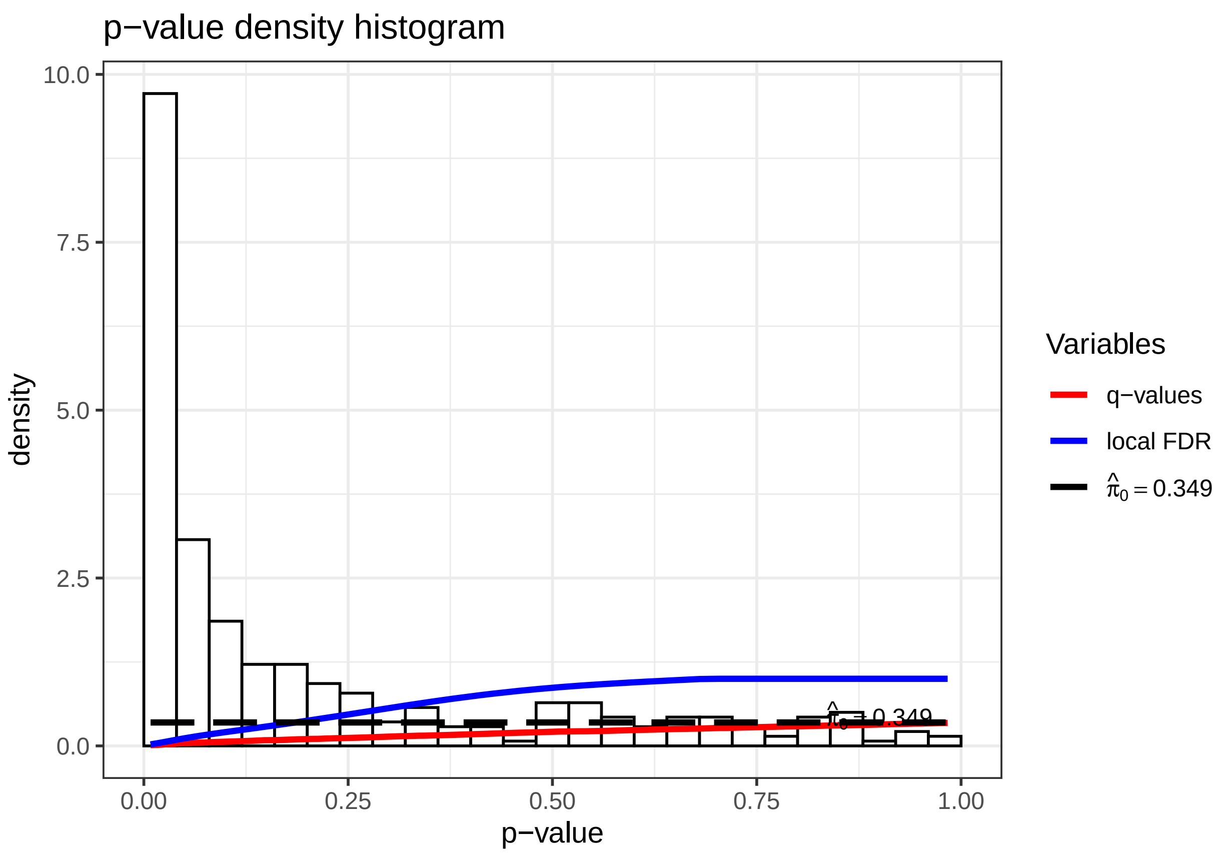
```
complexFeatures <- findComplexFeatures(traces=protTraces,
                                       complex_hypothesis = corumTargetsPlusDecoys)
```

After calculating FDR we can filter the data to retain only significant complexes

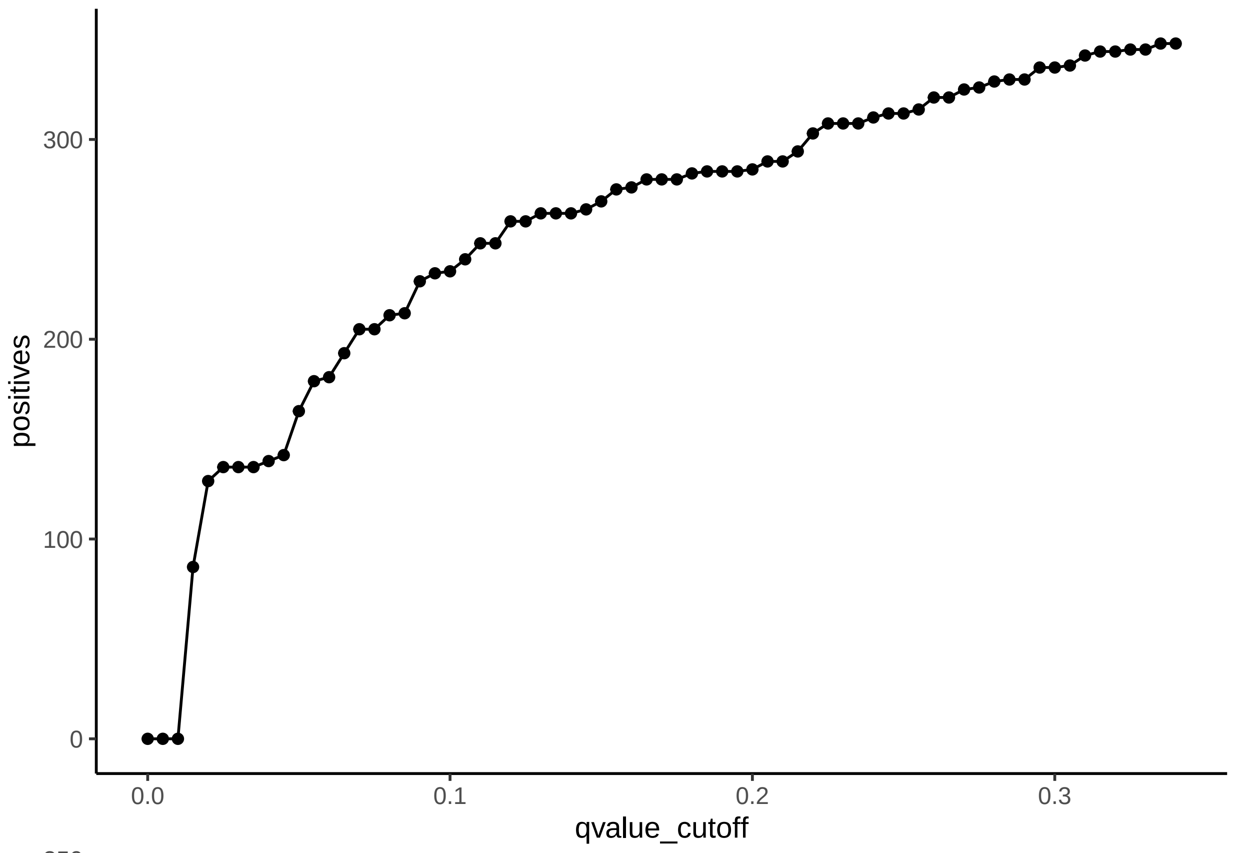
```
complexFeaturesScored <- calculateCoelutionScore(complexFeatures)
qvalueComplexFeaturesScored <- calculateQvalue(complexFeaturesScored)
```

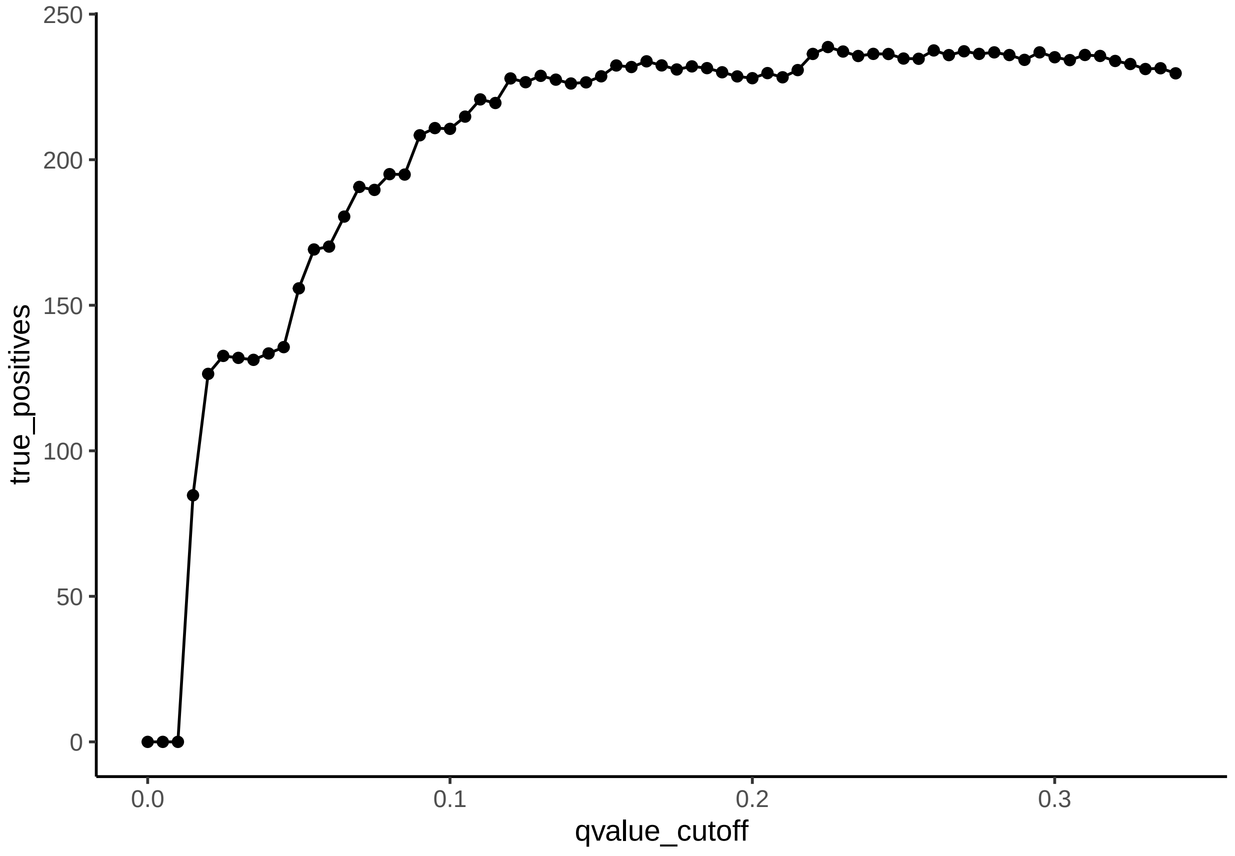






```r
qvalueComplexFeaturesScoredStats <- qvaluePositivesPlot(qvalueComplexFeaturesScored)
```


positives
300
200
100
0
0.0
0.1
0.2
0.3
qvalue_cutoff

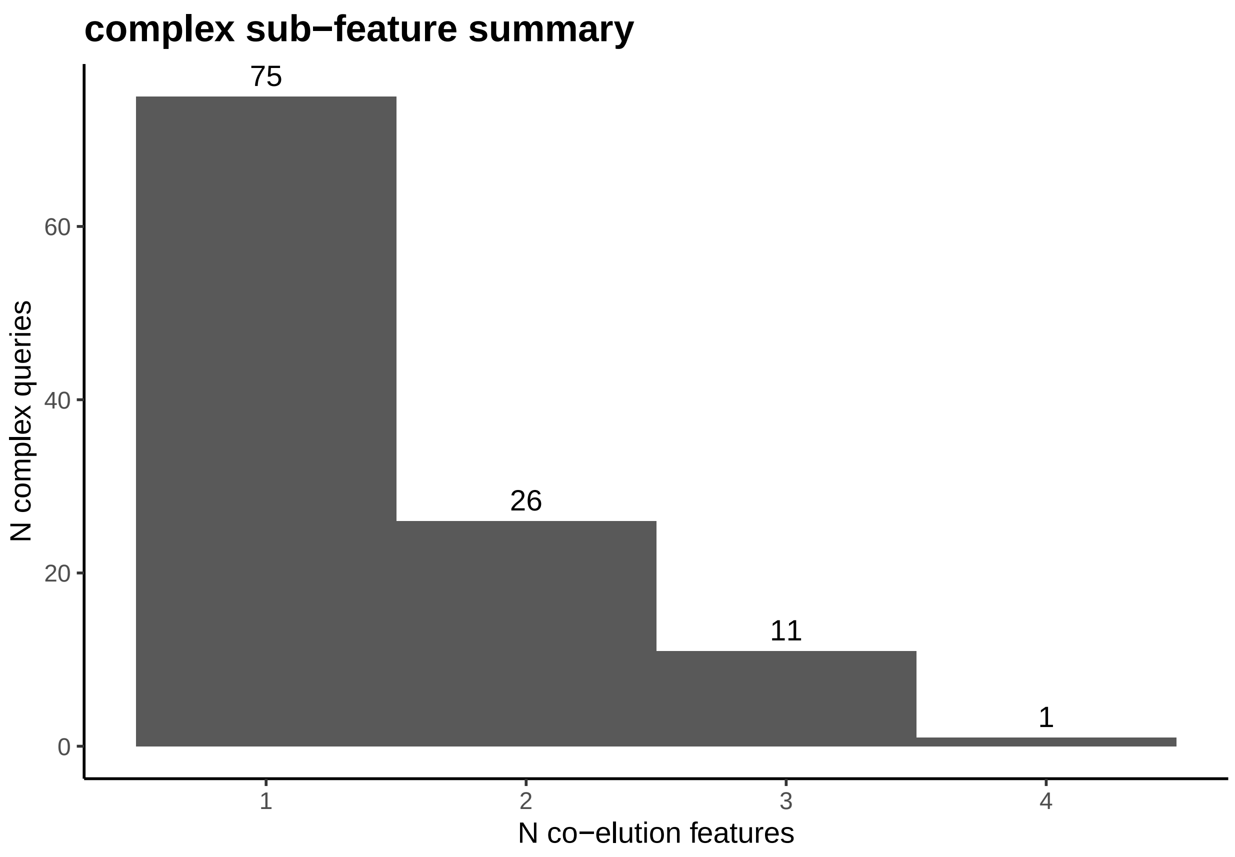

true_positives
250
200
150
100
50
0
0.0
0.1
0.2
0.3
qvalue_cutoff

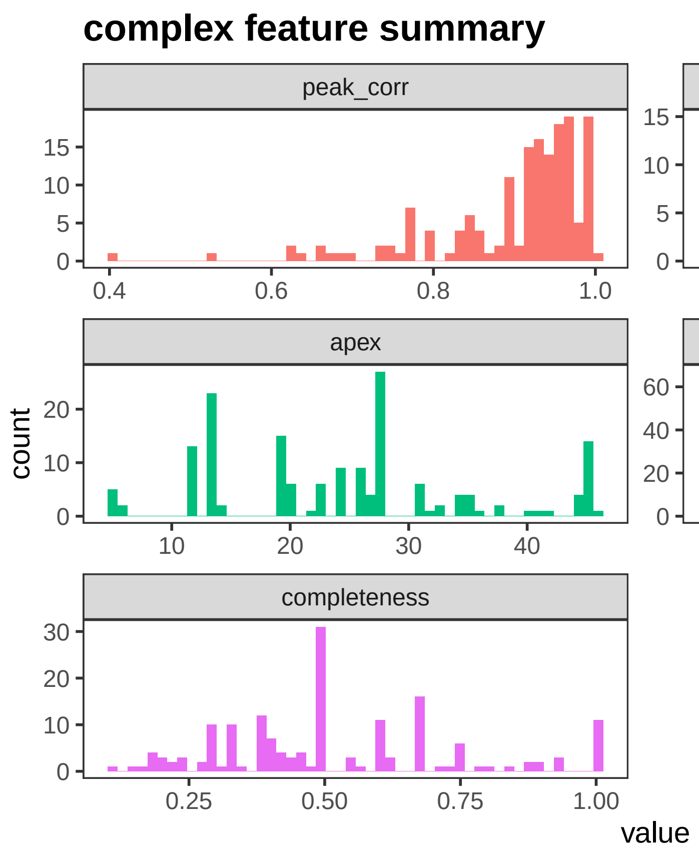
```
complexFeaturesFiltered <- subset(qvalueComplexFeaturesScored, qvalue <= 0.05)
```

plots and visualization

The summary function provides an overview of detected protein complexes

```
summarizeFeatures(complexFeaturesFiltered)
```



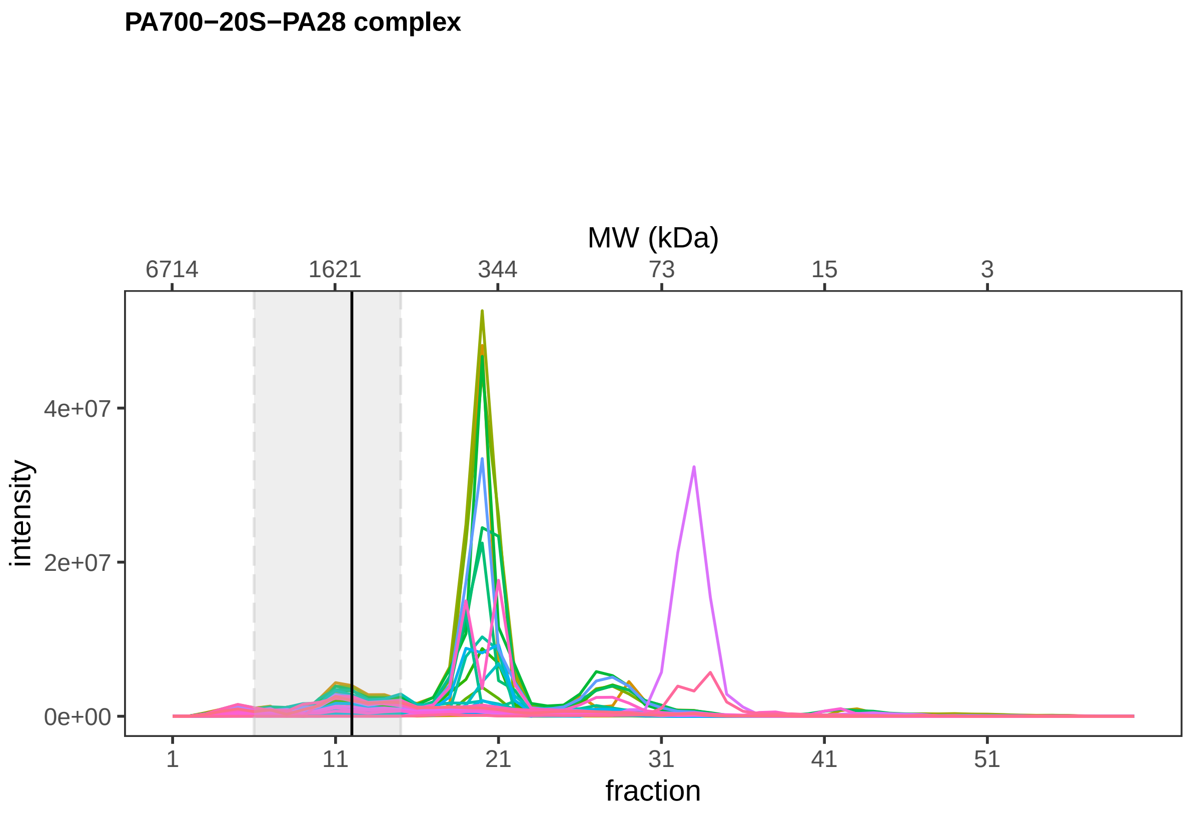


```
## $type
## [1] "complex"
##
## $totalFeatures
## [1] 164
##
## $totalConfirmedHypotheses
## [1] 113
##
## $totalHypothesesWithMultipleFeatures
## [1] 38
##
## $summaryFeatureCount
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.451   2.000   4.000
##
## $summaryCorrelation
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3985  0.8677  0.9342  0.8993  0.9629  0.9984
##
## $summaryArea
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 6.087e+05 1.365e+07 3.442e+07 1.645e+08 9.386e+07 1.481e+09
##
## $summaryNsubunitsAnnotated
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   4.000   6.500   8.707  12.000  44.000
```

```
##
## $summaryNsubunitsWithSignal
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   3.000   5.000   6.134   7.000  36.000
##
## $summaryNsubunitsDetected
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   3.000   4.329   5.000  32.000
##
## $summaryCompleteness
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1053  0.3750  0.5000  0.5211  0.6667  1.0000
```

CCprofiler allows fast plotting of all protein complexes with built in functions

```
# plot 26S proteasome
plotFeatures(feature_table = complexFeaturesFiltered,
             traces = protTraces,
             feature_id = "193",
             annotation_label="Entry_name",
             calibration = calibration_new,
             peak_area = TRUE)
```



References

Heusel, M., I. Bludau, G. Rosenberger, R. Hafen, M. Frank, A. Banaei-Esfahani, A. van Drogen, B. C. Collins, M. Gstaiger, and R. Aebersold. 2019. ‘Complex-centric proteome profiling by SEC-SWATH-MS’, Molecular Systems Biology, 15.

Rosenberger, G., I. Bludau, U. Schmitt, M. Heusel, C. L. Hunter, Y. S. Liu, M. J. MacCoss, B. X. MacLean, A. I. Nesvizhskii, P. G. A. Pedrioli, L. Reiter, H. L. Rost, S. Tate, Y. S. Ting, B. C. Collins, and R. Aebersold. 2017. 'Statistical control of peptide and protein error rates in large-scale targeted data-independent acquisition analyses', Nature Methods, 14: 921-+.

Rost, H. L., Y. Liu, G. D'Agostino, M. Zanella, P. Navarro, G. Rosenberger, B. C. Collins, L. Gillet, G. Testa, L. Malmstrom, and R. Aebersold. 2016. 'TRIC: an automated alignment strategy for reproducible protein quantification in targeted proteomics', Nat Methods, 13: 777-83.

Rost, H. L., G. Rosenberger, P. Navarro, L. Gillet, S. M. Miladinovic, O. T. Schubert, W. Wolskit, B. C. Collins, J. Malmstrom, L. Malmstrom, and R. Aebersold. 2014. 'OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data', Nature Biotechnology, 32: 219-23.