Proteoform annotation benchmark

Isabell Bludau

December 19th, 2018

Contents

Overview **1**

Load CCprofiler package, set working directory & load data **1**

Benchmark proteoform detection by minimum peptide correlation **2**

Benchmark proteoform separation by correlation clustering **3**

Benchmark proteoform resolved protein feature finding **4**

Overview

The goal of this workflow is to benchmark proteoform detection in CCprofiler. For this purpose, we perform an in silico generation of mixture proteins with multiple proteoforms by mixing peptides of different proteins. These mixture proteins are then used to evaluate the ability of our algorithm to: (a) correctly determine the mixture proteins, (b) correctly group the peptides of mixture proteins by their parental proteins and (c) detect mixture-resolved protein peak-groups

```
knitr::opts_chunk$set(eval = FALSE)
# rmarkdown::render("proteoformBenchmark.R", "pdf_document")
```

Load CCprofiler package, set working directory & load data

```
library(devtools)
library(nFactors)
if (length(grep("nas21.ethz.ch",getwd()))>0) {
  setwd("~/mysonas/CCprofiler")
  load_all()
  setwd("~/mysonas/PRPF8/analysis/output/benchmark")
  knitr::opts_knit$set(root.dir = '~/mysonas/PRPF8/analysis/output/benchmark',
                       echo = TRUE, eval=FALSE)
} else {
  setwd("/Volumes/ibludau-1/CCprofiler")
  load_all()
  setwd("/Volumes/ibludau-1/PRPF8/analysis/output/benchmark")
  knitr::opts_knit$set(root.dir = '/Volumes/ibludau-1/PRPF8/analysis/output/benchmark',
                       echo = TRUE, eval=FALSE)
}
# install_github("CCprofiler/CCprofiler", ref = "DA_module")
# library(CCprofiler)
```

Source functions required for benchmarking

```
source("../../CCprofilerAnalysis/benchmark/generateMixtureTraces.R")
source("../../CCprofilerAnalysis/benchmark/resolveProteoformSpecificFeatures.R")
```

Load data

```
peptide_traces <- readRDS("../traces_maxCorr_multi.rda")
calibrationFunctions <- readRDS("../calibration.rds")
#trueInteractions <- readRDS("string_binaryHypotheses.rds")
#trueInteractions[,idx:=.I]
#trueInteractions[,id:= paste(sort(c(a,b)),collapse = ";"), by=idx]
```

Benchmark proteoform detection by minimum peptide correlation

To benchmark the sensitivity of detecting genes with multiple proteoforms we generate 1000 mixture traces (peptide_traces_mixed) and append them to the normal peptide traces (negative_traces). The resulting combined traces set (benchmark_traces) is subsequently used for proteoform detection via the minimum correlation criterion. The recovery of true mixture proteins is evaluated across different adj. p-value cutoffs.

We iterate across mixing peptides from 2-4 parental proteins. For mixing, only 50% (or minimally 2) peptides are selected per parental protein.

```
summaryStats_MinCorr <- data.table()
for (i in c(2,3,4)) {
  peptide_traces_mixed <- generateMixtureTraces(peptide_traces,
                                                trueInteractions = NULL,
                                                n_proteins = i,
                                                n_mixtureTraces = 1000,
                                                peptide_ratio = 0.5,
                                                min_peptide_count = 2,
                                                seed = 123)

  negative_traces <- copy(peptide_traces)
  negative_traces$trace_annotation[,n_mixed_proteins := 1]
  negative_traces$trace_annotation[,is_mixed:=FALSE]

  benchmark_traces <- copy(negative_traces)
  benchmark_traces$traces <- rbind(negative_traces$traces, peptide_traces_mixed$traces)
  benchmark_traces$trace_annotation <- rbind(negative_traces$trace_annotation,
                                             peptide_traces_mixed$trace_annotation)

  benchmark_traces_minCorr <- calculateMinCorr(benchmark_traces,
                                               plot = T, PDF=T,
                                               name=paste0("minCorrHist_",i,
                                                           "mixedProteins"))

  benchmark_traces_minCorr_pval <- estimateProteoformPval(benchmark_traces_minCorr,
                                                          plot = T, PDF=T,
                                                          name=paste0("SplicePval_",i,
                                                                      "mixedProteins"))

  benchmark_stats <- evaluateProteoformSensitivityOfMixtureTraces(
    benchmark_traces_minCorr_pval,
    plot = T, PDF=T,
```

```
    name=paste0("TPR_vs_proteoform_pval_adj_",i,"mixedProteins"))

  summaryStats_MinCorr <- rbind(summaryStats_MinCorr,
                                benchmark_stats[proteoform_pval_adj==0.05])
  }

summaryStats_MinCorr
write.table(summaryStats_MinCorr,"summaryStats_mixedProtein_MinCorr.txt",
            quote=F, col.names = T, row.names = F)
```

Benchmark proteoform separation by correlation clustering

To benchmark the ability of our algorithm to correctly separate proteoforms based on peptide correlation clustering, we first filter the original peptide traces for proteins containing only one proteoform based on the minimum correlation criterion (adj. p-value > 0.05). The resulting traces (peptide_traces_highCorr) are used to generate 1000 mixture traces (peptide_traces_highCorr_mixed). The mixture traces are clustered and proteoforms are assigned based on a cluster height cutoff of 0.6. The clustering is evaluated by testing how many clusters contain peptides from more than one parental protein.

We iterate across mixing peptides from 2-4 parental proteins. For mixing, only 50% (or minimally 2) peptides are selected per parental protein.

```
peptide_traces_minCorr <- calculateMinCorr(peptide_traces,
                                           plot = T, PDF=T)

peptide_traces_minCorr_pval <- estimateProteoformPval(peptide_traces_minCorr,
                                                      plot = T, PDF=T)

proteins_highCorr <- unique(subset(peptide_traces_minCorr_pval$trace_annotation,
                                   proteoform_pval_adj > 0.05)$protein_id)
peptide_traces_highCorr <- subset(peptide_traces, proteins_highCorr,
                                  trace_subset_type = "protein_id")

summaryStats_mixedProteinClustering <- data.table()
for (i in c(2,3,4)) {
  peptide_traces_highCorr_mixed <- generateMixtureTraces(peptide_traces_highCorr,
                                                         trueInteractions = NULL,
                                                         n_proteins = i,
                                                         n_mixtureTraces = 1000,
                                                         peptide_ratio = 0.5,
                                                         min_peptide_count = 2,
                                                         seed = 123)

  peptide_traces_highCorr_mixed_clustered <- clusterPeptides(
    peptide_traces_highCorr_mixed,
    clusterN = NULL,
    clusterH = 0.6,
    nFactorAnalysis = F,
    pvclust=F,
    plot = T,
    PDF=T,
    name=paste0("hclust_",i,"mixedProteins_highCorr"))
```

```
  cluster_stats <- evaluateProteoformClusteringOfMixtureTraces(
    peptide_traces_highCorr_mixed_clustered)

  n_proteins <- nrow(cluster_stats)
  n_proteins_noMixedClusters <- nrow(cluster_stats[n_mistakes==0])
  fraction_noMixedClusters <- n_proteins_noMixedClusters/n_proteins
  n_proteins_correctClusterNumber <- nrow(cluster_stats[n_proteoforms == i])
  n_proteins_tooHighClusterNumber <- nrow(cluster_stats[n_proteoforms > i])
  n_proteins_tooLowClusterNumber <- nrow(cluster_stats[n_proteoforms < i])
  fraction_proteins_correctClusterNumber <- n_proteins_correctClusterNumber/n_proteins
  fraction_proteins_tooHighClusterNumber <- n_proteins_tooHighClusterNumber/n_proteins
  fraction_proteins_tooLowClusterNumber <- n_proteins_tooLowClusterNumber/n_proteins

  summaryStats_mixedProteinClustering <- rbind(
    summaryStats_mixedProteinClustering,
    data.table(n_mixedProteins=i,
    n_proteins=n_proteins,
    n_proteins_noMixedClusters=n_proteins_noMixedClusters,
    fraction_noMixedClusters=fraction_noMixedClusters,
    n_proteins_correctClusterNumber=n_proteins_correctClusterNumber,
    fraction_proteins_correctClusterNumber=fraction_proteins_correctClusterNumber,
    n_proteins_tooHighClusterNumber=n_proteins_tooHighClusterNumber,
    fraction_proteins_tooHighClusterNumber=fraction_proteins_tooHighClusterNumber,
    n_proteins_tooLowClusterNumber=n_proteins_tooLowClusterNumber,
    fraction_proteins_tooLowClusterNumber=fraction_proteins_tooLowClusterNumber))
  }

summaryStats_mixedProteinClustering
write.table(summaryStats_mixedProteinClustering,
            "summaryStats_mixedProteinClustering.txt",
            quote=F, col.names = T, row.names = F)
```

Benchmark proteoform resolved protein feature finding

To benchmark the ability of our algorithm to correctly separate proteoforms by proteoform resolved protein feature finding, we again use the original peptide traces filtered for proteins containing only one proteoform based on the minimum correlation criterion (adj. p-value > 0.05). The resulting traces (peptide_traces_highCorr) are used to generate 50 mixture traces (peptide_traces_highCorr_mixed). For the feature finding we assume to have already assigned proteoforms by the correlation based clustering, or in the benchmarking case by knowing the parental origin of a peptide in the mixture. The mixture traces are used for protein feature finding and subsequent proteoform-informed resolving of protein features into proteoform-specific features. During the resolving of proteoform specific features, the completeness is adjusted accroding to the proteoforms talking part in the feature. The proteoform resolution removes proteoforms within a mixed feature if the highest abundant peptide of the proteoform is <10% of the highest abundant peptide of the most abundant proteoform. The proteoform-resolved feature finding is evaluated by testing how many features contain peptides from more than one parental protein / more than one proteoform.

We iterate across mixing peptides from 2-4 parental proteins. For mixing, only 50% (or minimally 2) peptides are selected per parental protein.

```
summaryStatsFeatures <- data.table()
for (i in c(2,3,4)) {
  peptide_traces_highCorr_mixed <- generateMixtureTraces(peptide_traces_highCorr,
```

```
                                                    trueInteractions = NULL,
                                                    n_proteins = i,
                                                    n_mixtureTraces = 50,
                                                    peptide_ratio = 0.5,
                                                    min_peptide_count = 2,
                                                    seed = 123)

peptide_traces_highCorr_mixed_definedProteoforms <- copy(peptide_traces_highCorr_mixed)
peptide_traces_highCorr_mixed_definedProteoforms$trace_annotation[,proteoform_id:=
                                                                    uniprot_swissprot]

proteinFeatures  <- findProteinFeatures(
  traces=peptide_traces_highCorr_mixed_definedProteoforms,
  corr_cutoff=0.9,
  window_size=7,
  parallelized=F,
  n_cores=1,
  collapse_method="apex_only",
  perturb_cutoff= "5%",
  rt_height=1,
  smoothing_length=7,
  useRandomDecoyModel=T,
  quantLevel = "protein_id")
saveRDS(proteinFeatures, paste0("proteinFeatures_",i,"mixedProteins.rda"))

proteoformFeaturesResolved <- resolveProteoformSpecificFeatures(
  features=proteinFeatures,
  traces=peptide_traces_highCorr_mixed_definedProteoforms,
  minProteoformIntensityRatio=0.1,
  perturb_cutoff="5%")

filteredDataProteoformResolved <- scoreFeatures(
  proteoformFeaturesResolved,
  FDR=0.05, PDF=T,
  name=paste0("qvalueStats_proteoformFeaturesResolved_",i,"mixedProteins"))

saveRDS(filteredDataProteoformResolved,
        paste0("filteredDataProteoformResolved_",i,"mixedProteins.rda"))

pdf(paste0("mixedFeatures_proteoformResolved_",i,"mixedProteins.pdf"),width=4,height=4)
for (id in unique(filteredDataProteoformResolved$protein_id)){
  plotFeatures(feature_table = filteredDataProteoformResolved,
               traces = peptide_traces_highCorr_mixed_definedProteoforms,
               calibration=calibrationFunctions,
               feature_id = id,
               annotation_label="Entry_name",
               colour_by="Entry_name",
               peak_area = T,
               legend = F,
               onlyBest = F)
}
dev.off()
```

```
  n_proteins <- length(unique(filteredDataProteoformResolved$protein_id))
  n_protein_features <- nrow(filteredDataProteoformResolved)
  n_protein_features_singleProteoform <-
    nrow(filteredDataProteoformResolved[n_proteoform_ids==1])
  n_protein_features_multipleProteoforms <-
    nrow(filteredDataProteoformResolved[n_proteoform_ids>1])
  fraction_protein_features_singleProteoform <-
    n_protein_features_singleProteoform/n_protein_features
  fraction_protein_features_multipleProteoforms <-
    n_protein_features_multipleProteoforms/n_protein_features

  summaryStatsFeatures <- rbind(summaryStatsFeatures, data.table(
    n_mixedProteins = i,
    n_proteins = n_proteins,
    n_protein_features = n_protein_features,
    n_protein_features_singleProteoform =
      n_protein_features_singleProteoform,
    fraction_protein_features_singleProteoform =
      fraction_protein_features_singleProteoform,
    n_protein_features_multipleProteoforms =
      n_protein_features_multipleProteoforms,
    fraction_protein_features_multipleProteoforms =
      fraction_protein_features_multipleProteoforms))
}

summaryStatsFeatures
write.table(summaryStatsFeatures,"summaryStats_mixedProteinFeatureFinding.txt",
            quote=F, col.names = T, row.names = F)
```