How to generate an annotation\_features.txt file

Miguel Cosenza

14 October, 2021

Table of Contents

[1 Example based on annotated semi-specific peptides 1](#_Toc85108330)

[1.1 First load the file of annotated peptides 1](#_Toc85108331)

[1.2 Then select the interesting feature 2](#_Toc85108332)

[2 Example based on protein names from headers 3](#_Toc85108333)

In this document I am including a small example on how to generate a annotation\_features.txt file to be used as an input to do ‘selected’ FDR control after limma on a interestin features.

**Note**: the outputs that can be generated in this example would not match the ones in the sample data. This document is intender to provide a guide on how to approach such a problem of creating the annotation\_features.txt. You need to adapt it so the IDs in your annotation\_features.txt file would match the ones in your input\_limma.txt file.

# 1 Example based on annotated semi-specific peptides

We already have a script that allows us to annotate our peptides regarding their specificity ([Mapping peptides to proteins in FASTA - Annotate](https://github.com/MiguelCos/mapping_peptides_to_proteins_from_fasta_file)) and we can use that to create our annotation\_features.txt file.

## 1.1 First load the file of annotated peptides

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.2 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.1 v forcats 0.5.1

## Warning: package 'readr' was built under R version 4.1.1

## Warning: package 'dplyr' was built under R version 4.1.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(here)

## here() starts at C:/Users/migue/OneDrive/Documentos/R\_Projects/7\_scripts\_workflows/limma\_FDR\_on\_feature\_subset

annot\_peptides <- read\_tsv(here("results/peptide\_annotation\_pif75\_32searchv2.tsv"))

## Rows: 78566 Columns: 14

## -- Column specification --------------------------------------------------------  
## Delimiter: "\t"  
## chr (11): protein\_id, protein\_description, Peptide, last\_aa, aa\_after, aa\_be...  
## dbl (3): protein\_length, start\_position, end\_position

##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

## 1.2 Then select the interesting feature

Take a look at the names of the columns

names(annot\_peptides)

## [1] "protein\_id" "protein\_description" "protein\_length"   
## [4] "Peptide" "start\_position" "end\_position"   
## [7] "last\_aa" "aa\_after" "aa\_before"   
## [10] "following\_10\_resid" "previous\_10\_resid" "previous\_all\_resid"   
## [13] "semi\_type" "specificity"

In this case, the columns related to the unique ID of the Peptide + Protein (index) and the type of specificity.

**First we create our index variable so it matches the one in the input\_limma.txt file.**

annot\_peptides\_2 <- mutate(annot\_peptides, # annotated peptides table as input  
 index = paste(protein\_id, Peptide, "\_")) # new variable index is created by concatenating protein ID and peptide.

**Then we select the interesting columns**

In this case: index and semi\_type (containing the specificity information per peptide/feature)

annotation\_features <- dplyr::select(annot\_peptides\_2,  
 ID = index, # changed index to ID column name  
 feature\_category = semi\_type) # changed semi\_type to feature\_category column name

We can now save the annotation\_features.txt file

write\_delim(annotation\_features,  
 file = here("sample/data/annotation\_features\_semi\_try.txt"),   
 delim = "\t")

# 2 Example based on protein names from headers

In the case of metaproteomics studies, we might be interested in labelling proteins as ‘human’ or ‘non-human/bacteria’.

If we have the complete Uniprot identifier of the proteins, we can do that annotation easily.

We load a sample dataset containing protein headers/IDs in the first column

proteins\_w\_headers <- read\_tsv(here("sample/data/sample\_file\_ids\_header.tsv"))

## Rows: 3029 Columns: 7

## -- Column specification --------------------------------------------------------  
## Delimiter: "\t"  
## chr (1): header\_id  
## dbl (6): DOTH41b\_SF\_C, DOTH42b\_SF\_C, DOTH43b\_SF\_C, DOTH44b\_SF\_C, DOTH45b\_SF\_...

##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

If we check the first 6 rows, we see that the first column contain the protein headers names from the fasta file. We can use that to know if it comes from bacteria or human.

head(proteins\_w\_headers)

## # A tibble: 6 x 7  
## header\_id DOTH41b\_SF\_C DOTH42b\_SF\_C DOTH43b\_SF\_C DOTH44b\_SF\_C DOTH45b\_SF\_C  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 >sp|A0A075B6~ 36138. 42161 NA 214431 563395  
## 2 >sp|A0A075B6~ 13406900 14760200 2048760 NA 864503  
## 3 >sp|A0A075B6~ NA NA NA NA 165199  
## 4 >tr|A0A075B7~ 1434720 823523 NA 608046 240933  
## 5 >tr|A0A075B7~ 40787000 22400700 12236400 2292910 5819930  
## 6 >sp|A0A087WS~ 2666320 2868430 NA 3204700 3568570  
## # ... with 1 more variable: DOTH46b\_SF\_C <dbl>

Now we annotate the proteins as bacterial or human

proteins\_w\_annotation <- mutate(proteins\_w\_headers, # input object  
 feature\_category = if\_else(str\_detect(header\_id,"\_HUMAN"),   
 true = "human",  
 false = "bacteria")) # feature category is the name that this column should have in the `annotation\_features.txt`

If we look at the last two columns, we see that, for each protein, we have a feature annotation.

proteins\_w\_annotation[24:33,c(5:8)]

## # A tibble: 10 x 4  
## DOTH44b\_SF\_C DOTH45b\_SF\_C DOTH46b\_SF\_C feature\_category  
## <dbl> <dbl> <dbl> <chr>   
## 1 3967610 NA NA human   
## 2 NA NA NA human   
## 3 6140610 6053250 7989480 human   
## 4 NA NA NA bacteria   
## 5 2073590 NA NA bacteria   
## 6 NA NA NA bacteria   
## 7 231245 NA NA bacteria   
## 8 NA NA NA bacteria   
## 9 1639280 NA NA bacteria   
## 10 7970010 NA NA bacteria

We can now select the two interesting columns (ID and feature\_category) and save them in a file.

annotation\_features2 <- dplyr::select(proteins\_w\_annotation,  
 ID = header\_id, # changed index to ID column name  
 feature\_category) # changed semi\_type to feature\_category column name

write\_delim(annotation\_features2,  
 file = here("sample/data/annotation\_features\_metaprot.txt"),   
 delim = "\t")