Label ROC Data

Michael A. Gilchrist

24 Jul 2020

Preliminary Information

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

To compile in emacs use M-n e

Purpose

Use information from FASTA file used to fit ROC to add isoform.ID and WormBase.ID information

Load Libraries

```
library(Biostrings) ## process first to avoid conflicts
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which, which.max, which.min
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
##
## Attaching package: 'S4Vectors'
## The following object is masked from 'package:base':
##
##     expand.grid
## Loading required package: IRanges
## Loading required package: XVector
##
## Attaching package: 'Biostrings'
## The following object is masked from 'package:base':
##
##     strsplit
```

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
## The following object is masked from 'package:S4Vectors':
##
##     expand
```

```
library(tibble)
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:Biostrings':
##
##     collapse, intersect, setdiff, setequal, union
## The following object is masked from 'package:XVector':
##
##     slice
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
```

```
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(forcats)
library(ggplot2)
library(knitr)
library(ggpubr)
library(ggpmisc)
library(purrr)
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:XVector':
##
##     compact
```

```
## The following object is masked from 'package:IRanges':
##
##     reduce
```

```
creationInfo <- paste0("\tDate: ", date(), "\n\tLocation: ", sub(".*/AcrossTissue", "AcrossTissue", getw

exportData=TRUE ## Flag for running save(), write.csv() and other output commands
```

Load and Shape Data

ROC

Load FASTA to get isoform and WormBase IDs (names)

```
## Load WB FASTA file
## Using Biostrings function which is not a standard df
seqData <- readDNAStringSet("Input/c_elegans.PRJNA13758.WS277.CDS_transcripts.fa")

seqLength <- width(seqData)

## names are really long descriptions.
## NEed to extract relevant part
seqDesc  <- as_tibble_col(x=names(seqData), column_name = "desc")
seqID <- separate(data = seqDesc, col = desc, into=c("isoform.ID", "WormBase.ID"), sep=" gene=", remove=
rm(seqDesc)

## sub example
##WormBase.ID <- sub(".* gene=([WBGEne0-9]+)", "\\1", seqDesc)



## verify there's a match for each entry
if(sum(is.na(seqID$WormBase.ID)) ==0) print("Every entry matches")
```

```
## [1] "Every entry matches"
```

```
## Verify that all 'names' are unique.
if(length(seqID$WormBase.ID) != length(unique(seqID$WormBase.ID))) print("Some WormBase IDs appear twic
```

```
## [1] "Some WormBase IDs appear twice due to isoforms"
```

```
if(length(seqID$isoform.ID) == length(unique(seqID$isoform.ID))) print("Every isoform.ID is unique as e
```

```
## [1] "Every isoform.ID is unique as expected."
```

Label ROC estimates

Load ROC Estimates and bind isoform.ID and WormBase.ID with Phi Values

```
## Import Phi Values from ROC Output

## detailed information on phi: posterior mean, posterior mean of log10(phi), etc
## StdError really StdDev of posterior
## This will change with a ROC update
unlabeledROCOutput <-
    readr::read_csv("Input/ROC_unlabeled.phi.summaries.with.sphi.equal.2.8.csv")
```

```
## Parsed with column specification:
## cols(
##   PHI = col_double(),
##   log10.PHI = col_double(),
##   Std.Dev = col_double(),
##   log10.Std.Dev = col_double(),
##   `0.025` = col_double(),
##   `0.975` = col_double(),
##   log10.0.025 = col_double(),
##   log10.0.975 = col_double()
## )
```

```
labeledROCOutput <- bind_cols(seqID, unlabeledROCOutput, seqLength)
```

```
## New names:
## * NA -> ...11
```

```
write.csv(x=labeledROCOutput, file="Output/ROC_labeled.phi.summaries.with.sphi.equal.2.8.csv", quote=FA
```