Phosphomotifs_toy_example

David Bradley

16 March 2019

Calculation of phosphorylation motif enrichments for four alveolate species

In the manuscript, phosphorylation motif enrichment p-values are represented in Figure 4 for several different species (columns) and phosphorylation motifs (rows). We illustrate the code used to generate these results by repeating the analysis just for the four alveolate species (*Plasmodium falciparum*, *Plasmodium berghei*, *Toxoplasma gondii*, and *Tetrahymena thermophila*) that were included in the dataset.

The first step is to load phosphoproteome data for each of the four species into R:

```
# Load in the background data and foreground data

bg_list <- readRDS('bg_list_shuffle_x10.rds')
fg_list <- readRDS('fg_list_publication.rds')

bg_list <- bg_list[names(fg_list) %in% c("falciparum","berghei","gondii","Tthermophila")]
fg_list <- fg_list[names(fg_list) %in% c("falciparum","berghei","gondii","Tthermophila")]
```

The number of unique phosphorylation sites known in each of the four species (*Plasmodium falciparum*, *Plasmodium berghei*, *Toxoplasma gondii*, and *Tetrahymena thermophila*) is as follows:

```
# Load in the foreground data and background data

print(length(fg_list[[1]]))
```

```
## [1] 11588
```

```
print(length(fg_list[[2]]))
```

```
## [1] 9458
```

```
print(length(fg_list[[3]]))
```

```
## [1] 28846
```

```
print(length(fg_list[[4]]))
```

```
## [1] 1934
```

Now, for each of the four species, we use an R-based implementation of 'motif-x' to extract all significant motifs. We use the default parameters of motif-x for this purpose (binomial p-value less than 1e-6, and at least 20 foreground occurrences of the motif required).

```
# Find significant phosphorylation motifs for each of the four species

library(rmotifx)

mot_list <- NULL

for (i in 1:length(fg_list)) {

  print(i)
```

```
  mot1 <- motifx(unique(unlist(fg_list[i])), unique(unlist(bg_list[i])),
                 central.res = 'ST', min.seqs = 20, pval.cutoff = 1e-6)

  mot_list <- c(mot_list,list(mot1[,1]))

}
```

A list of all significant phosphorylation motifs found for *Plasmodium falciparum* is given below:

```
# Find significant phosphorylation motifs for each of the four species

print(mot_list[1])
```

```
## [[1]]
##  [1] "...KK..[ST]L......" "...KK..[ST]......." "....K..[ST].E....."
##  [4] "....K..[ST].D....." "...KR..[ST]......." "....RS.[ST]......."
##  [7] "....KR.[ST]......." ".K..S..[ST]......." ".......[ST]PS....."
## [10] ".......[ST]D.E...." "....KS.[ST]......." "....R..[ST].D....."
## [13] ".......[ST].E.I..." ".......[ST].DE...." ".....M.[ST].E....."
## [16] "....R..[ST].S....." ".......[ST].E....." "....K..[ST]F......"
## [19] ".......[ST]DD....." ".......[ST].D....." ".......[ST].S....."
## [22] ".......[ST]P.K...." "....R..[ST]L......" ".....Y.[ST]D......"
## [25] ".......[ST]P......" "......G[ST]......." "....R..[ST]......."
## [28] "....KK.[ST]......." "....S..[ST]L......" ".......[ST]D......"
## [31] ".....S.[ST]......." "....S..[ST]......." "....K..[ST]......."
## [34] ".......[ST]F......" ".......[ST]NNI...." ".......[ST].T....."
## [37] ".......[ST]S......" ".......[ST]..S...."
```

For this reduced example, let us consider only those motifs identified in at least three of the four species

```
# Tally the motifs from each species
motx_table <- table(unlist(mot_list))
motx_table_reduced <- motx_table[which(motx_table >= 3)]
print(motx_table_reduced)
```

```
##
## ...KK..[ST]....... ...KR..[ST]....... ....K..[ST].......
##                  3                  3                  3
## ....R..[ST]....... .......[ST].DE.... .......[ST]D.E....
##                  4                  3                  3
## .......[ST]P......
##                  4
```

As in the manuscript, 'K' and 'R' will be considered to be synonymous, as will 'D' and 'E'. Let's now calculate binomial pvalue for each of these motifs. The code for two motifs will be given below as examples; the other three will be hidden for brevity, as only the amino acid symbol and position changes:

```
# Find significant phosphorylation motifs for each of the four species

# D/E +1 and D/E + 3

binom_vec <- NULL

for (i in 1:length(fg_list)) {

  fg <- unique(fg_list[[i]])
  bg <- unique(bg_list[[i]])
```

```
  fg <- fg[rapply(strsplit(fg,split=''), function(x) x[8] %in% c('S','T'))]

  # Find the total number of foreground peptides and background
  # peptides with the D/E+1 signature

  fg_new <- fg[rapply(strsplit(fg,split=''), function(x) x[9] %in% c('D','E'))]
  bg_new <- bg[rapply(strsplit(bg,split=''), function(x) x[9] %in% c('D','E'))]

  # Now find the total number of foreground peptides with the S/T-D/E-x-D/E signature

  fg_count <- length(fg[rapply(strsplit(fg,split=''),
                               function(x) x[9] %in% c('D','E') & x[11] %in% c('D','E'))])

  # In the background, find the fraction of D/E+1
  # peptides that also have the S/T-D/E-x-D/E motif. This is
  # our 'null expectation' for the proportion of
  # foreground D/E+1 peptides with the S/T-D/E-x-D/E motif.

  pos_bg <- length(bg[rapply(strsplit(bg,split=''), function(x)
    x[9] %in% c('D','E') & x[11] %in% c('D','E'))])/length(bg_new)

  # We now have sufficient information to calculate binomial p-values

  pval <- 1-pbinom(fg_count-1,length(fg_new),pos_bg)

  binom_vec <- c(binom_vec,pval)

}

binom_vec_CK2 <- binom_vec

# P+1

# Repeat the process for the P+1 signature (same logic applies)

binom_vec <- NULL

for (i in 1:length(fg_list)) {

  fg <- fg_list[[i]]


  bg <- unique(bg_list[[i]])

  fg <- fg[rapply(strsplit(fg,split=''), function(x) x[8] %in% c('S','T'))]
  fg_count <- length(fg[rapply(strsplit(fg,split=''),
                               function(x) x[9] %in% c('P'))])
  pos_bg <- length(bg[rapply(strsplit(bg,split=''),
                             function(x) x[9] %in% c('P'))])/length(bg)

  pval <- 1-pbinom(fg_count-1,length(fg),pos_bg)

  binom_vec <- c(binom_vec,pval)
```

```
}

binom_vec_Pp1 <- binom_vec
```

Finally, we will collect the enrichment p-values into a matrix, and then use the matrix to generate a heatmap:

```
# Now generate a heatmap of the data

library(ComplexHeatmap)
```

```
## Loading required package: grid
```

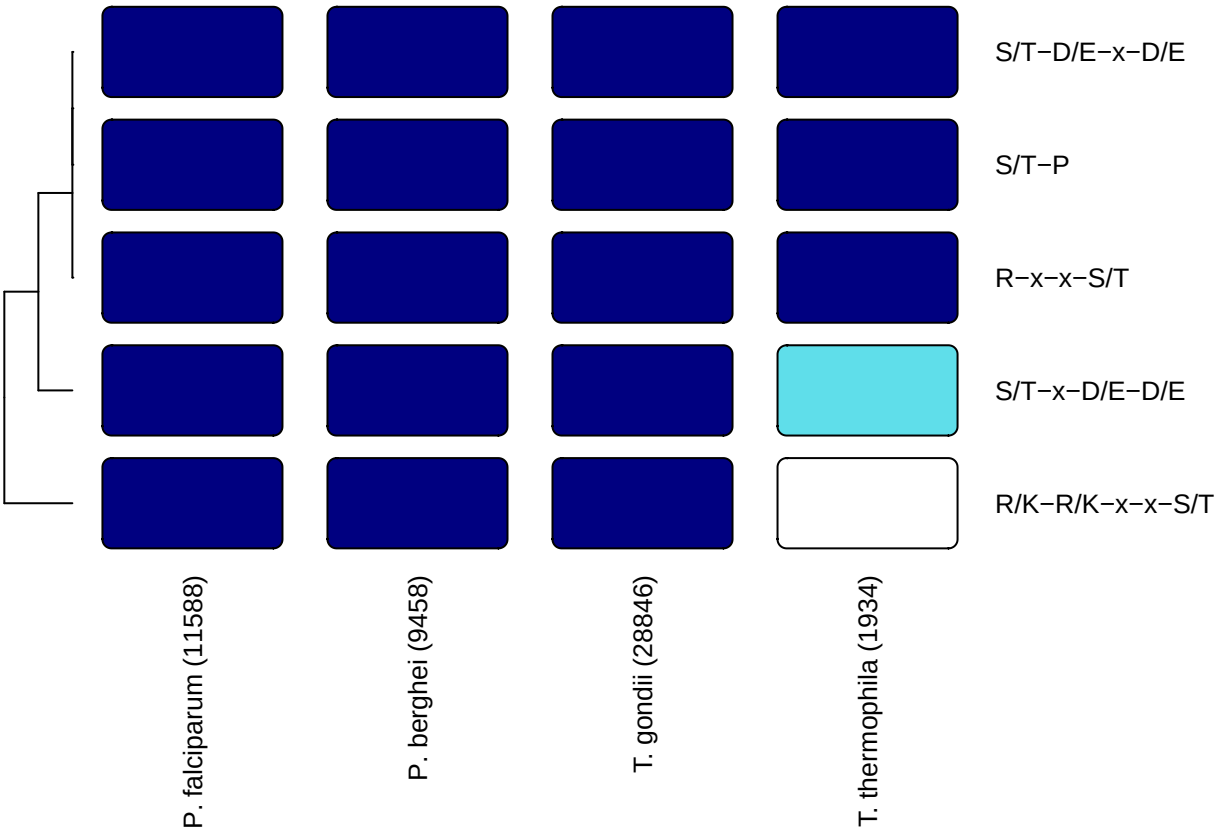
```
library(circlize)
```

```
## ========================================
## circlize version 0.4.3
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: http://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
## ========================================
```

```
species <- c('P. falciparum','P. berghei', 'T. gondii', 'T. thermophila')

mat2 <- rbind(binom_vec_CK2,binom_vec_Pp1,binom_vec_DEp2_DEp3,
              binom_vec_Rn3_Rn4,binom_vec_Rn3)

colnames(mat2) <- paste(species,' (',rapply(fg_list, function(x) length(x)),')',sep='')

rownames(mat2) <- c('S/T-D/E-x-D/E','S/T-P','S/T-x-D/E-D/E','R/K-R/K-x-x-S/T','R-x-x-S/T')

# Discretize the mat2 values

for (i in 1:nrow(mat2)) {
  for (j in 1:ncol(mat2)) {
    if(mat2[i,j] > 0.01) {mat2[i,j] = 0;next}
    if(mat2[i,j] <= 0.01 & mat2[i,j] >= 0.0001) {mat2[i,j] = 1;next}
    if(mat2[i,j] <= 0.001 & mat2[i,j] >= 0.000001) {mat2[i,j] = 2;next}
    if(mat2[i,j] <= 0.000001) {mat2[i,j] = 3;next}
  }
}


ht <- Heatmap(mat2, rect_gp = gpar(col = "white",
            lwd = 3, type = "none"),col=c('white',
            colors()[c(8)],colors()[c(121)],colors()[491]),
              cell_fun = function(j, i, x, y, width, height, fill) {
                grid.roundrect(x, y, width*0.8, height*0.80, gp = gpar(fill = fill))
              }
              ,cluster_rows = TRUE, cluster_columns = FALSE,
            row_names_side = NULL,row_names_gp = gpar(fontsize = 10),
              column_names_gp = gpar(fontsize = 10))
```

We now have sufficient data to generate the heatmap:

```
draw(ht, show_heatmap_legend = FALSE)
```



As detailed in the manuscript, the darkest blue colour is used for p-values below 1e-6, while the lighter blue is used for pvalues above 1e-6 but below 1e-4. White is used for p-values above 1e-2.