

PhosMap

Dongdong Zhan

2019-05-24

Introduction

PhosMap is a comprehensive R package for analyzing quantitative phosphoproteomics data, which provides multiple functions for users as follow: - (1) clustering: principal component analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE); - (2) differential expression analysis; - (3) time course analysis; - (4) kinase activity prediction to find activated/deactivated kinases from the kinase-substrate database - (5) phosphorylation motif enrichment analysis to provide clues for finding candidate kinases that are not present in the database; - (6) and data visualization.

Loading data

To test the efficacy of PhosMap and help users get started quickly, we collected a dataset including 39 phosphoproteomics and 32 proteomics raw files deposited in the ProteomeXchange Consortium (Ressa, et al., 2018). The partial key intermediate results were provided for users to master PhosMap. We have embedded intermediate results from demo into PhosMap for help users get started.

```
# Load PhosMap
library('PhosMap')
# Load intermediate results from https://github.com/ecnuzdd/PhosMap_datasets
ftp_url <- "ftp://111.198.139.72:4000/pub/PhosMap_datasets/BRAFi.RData"
load_data <- load_data_with_ftp(ftp_url, 'Rdata')

## First loading data from FTP sever, it may take a few minutes.

## Downloading data from ftp://111.198.139.72:4000/pub/PhosMap_datasets/BRAFi.RData.

## Completing the RData load.

writeBin(load_data, "BRAFi.RData")
load("BRAFi.RData")

# background_df A data frame for motif enrichment analysis as background
# combined_df_with_mapped_gene_symbol Get a data frame mapped GI number to Gene Symbol
# data_frame_normalization_with_control_no_pair A data frame containing phosphoproteomics data
# normalized by proteomics data.
# foreground_df A data frame for motif enrichment analysis as foreground.
# fuzzy_input_df A data frame for time course analysis as input.
# group A factor for experiment group information.
# merge_df_with_phospho_peptides A merged phosphoproteomics data frame based on peptides files (unique
# ID).
# motif_group_m_ratio_df_mat A matrix for motif profile.
# phospho_data_filtering_STY_and_normalization A phosphoproteomics data frame after normalization and
# filtering.
# profiling_data_normalized A proteomics data frame after normalization and filtering.
# summary_df_of_unique_proteins_with_sites A data frame that phosphorylation sites had been mapping to
# protein sequence and eliminated redundancy.
```

Data pre-processing

An intact data pre-processing procedure of phosphoproteomics data covered three main steps: merging input files after quality control, mapping phosphorylation sites (p-sites) to the corresponding protein sequence and data normalization.

Merging input files after quality control

'Phosphoproteomics data' and 'The phosphoproteomics experimental design' are required as input. For p-sites detected by Mascot, PhosMap could provide confidence probability of p-sites extracted from Mascot xml file. For p-sites detected by other software, a two column table including their corresponding sequences and confidence probability was indispensable. Then quality control at peptide and site levels for each experiment was performed.

```
BASE_DIR <- getwd() # working directory
BASE_DIR <- normalizePath(BASE_DIR)
phosphorylation_exp_design_info_file_path <- normalizePath(file.path(BASE_DIR,
'phosphorylation_exp_design_info.txt'))
phosphorylation_peptide_dir <- normalizePath(file.path(BASE_DIR, 'phosphorylation_peptide_txt'))

if(FALSE){
  # if you have xml files from mascot results, you can run the cmd to parser them to text files.
  mascot_xml_dir <- normalizePath(file.path(BASE_DIR, 'mascot_xml'))
```

```
mascot_txt_dir <- normalizePath(file.path(BASE_DIR, 'mascot_txt'))
extract_psites_score(phosphorylation_exp_design_info_file_path, mascot_xml_dir, mascot_txt_dir)

# Based on above-mentioned text files from Mascot results,
# the following cmd can generate CSV files of phosphorylation sites with confidence score.
psites_score_dir <- normalizePath(file.path(BASE_DIR, 'psites_score_txt'))
generate_psites_score_file(mascot_txt_dir, phosphorylation_peptide_dir, psites_score_dir)
}
# Merge phosphoproteomics data based on peptides files (unique ID).
# If qc = TRUE, considering confidence score of phosphorylation sites.
# A merged phosphoproteomics data frame based on peptides files (unique ID).
merge_df_with_phospho_peptides <- pre_process_filter_psites(
  phosphorylation_peptide_dir,
  psites_score_dir,
  phosphorylation_exp_design_info_file_path,
  qc = TRUE,
  min_score = 20,
  min_FDR = 0.01
)
```

Mapping phosphorylation sites (p-sites) to the corresponding protein sequence

- Mapping protein gi to gene symbol and outputing expression profile matrix with gene symnol.

```
# Get a data frame mapped GI number to Gene Symbol.
# system.time({
#   combined_df_with_mapped_gene_symbol = get_combined_data_frame(
#     merge_df_with_phospho_peptides, species = 'human', id_type = 'RefSeq_Protein_GI'
#   )
# })
head(combined_df_with_mapped_gene_symbol)
```

	Sequence		ID	Modification	GeneSymbol	Exp027012
## 1	AAAAAATAPPsPGPAQPGPR	gi 39930517	Phospho (ST)(11)		SAMD1	191101700
## 2	AAALSLSTLASPK	gi 4758248	Phospho (ST)(11)		EFNB1	19612020
## 3	AAALSLSTLASPK	gi 4758248	Phospho (ST)(5)		EFNB1	225996500
## 4	AAALSLSTLASPK	gi 4758248	Phospho (ST)(7)		EFNB1	25986220
## 5	AAAtPESQEPQAK	gi 13491174	Phospho (ST)(4)		MARCKSL1	543292900
## 6	AADPPAENSsAPEAQGGAE	gi 34098946	Phospho (ST)(10)		YBX1	385447600
##	Exp027013	Exp027014	Exp027015	Exp027016	Exp027017	Exp027018
## 1	72397330	0	32192620	231611600	124170700	28650550
## 2	17807950	19357340	79808370	84246970	56323030	48652040
## 3	148918100	77889940	360054600	226245100	189362500	203163300
## 4	12633960	0	0	0	0	0
## 5	1942228000	1524829000	1160647000	2877282000	1013506000	1597019000
## 6	624299300	579280200	364341700	649720200	415048000	890674400
##	Exp027019	Exp027020	Exp027021	Exp027022	Exp027023	Exp027024
## 1	31780240	123238200	30951010	0	159176400	0
## 2	33331610	0	26782070	59331840	81300420	11794550
## 3	159109000	112891900	117936700	130344300	136386600	40931780
## 4	0	0	0	0	0	0
## 5	578654800	2225267000	1101911000	2347455000	800039800	32572150
## 6	803288700	1327916000	637003800	1049919000	917176600	96717140
##	Exp027025	Exp027026	Exp027027	Exp027028	Exp027029	Exp027030
## 1	0	0	36968750	345538800	167434500	232864900
## 2	25698570	48053290	11260400	49733950	17026620	33251190
## 3	40848750	84373550	18245820	449307600	27093610	205768900
## 4	0	0	0	0	0	7733698
## 5	1403377000	2010127000	1768473000	2510754000	2389508000	2335473000
## 6	408221700	880458700	551895600	1435267000	626601400	653992300
##	Exp027031	Exp027032	Exp027033	Exp027034	Exp027035	Exp027036
## 1	0	341993300	171803500	0	38057890	109961100
## 2	68674110	82809850	20488140	66844210	94660320	26547410
## 3	140357500	203682300	71224800	170784100	153310800	41458280
## 4	0	5456458	0	0	0	0
## 5	2901491000	2740149000	1173034000	315909400	341922700	1432846000
## 6	693000900	766916700	568389600	639079000	754503200	1056575000
##	Exp027037	Exp027038	Exp027039	Exp027040	Exp027041	Exp027042
## 1	167525000	165389700	51471910	143915200	356015900	208103900
## 2	0	0	44516010	107605300	33192480	0
## 3	70873710	99639770	132810800	78469830	92153170	344426900
## 4	0	0	9818745	0	0	0
## 5	2104620000	2328424000	1673055000	2601683000	2413220000	1619502000
## 6	807748200	978603400	685710400	844008900	898072800	449702300
##	Exp027043	Exp027044	Exp027045	Exp027046	Exp027047	Exp027048
## 1	101188100	0	54273770	63307160	83332280	162916400
## 2	0	0	0	0	0	22553820
## 3	134677700	68013420	27048770	52237850	65281570	116332000
## 4	0	0	0	0	0	0
## 5	101506600	1102288000	865370300	172955100	192015000	687942500
## 6	267850800	806592800	494795000	0	315606800	295383200

```
## Exp027050
## 1 271443600
## 2 0
## 3 0
## 4 0
## 5 1428628000
## 6 0
```

- Constructing the data frame with unique phosphorylation site for each protein sequence.

```
# Assign psites to protein sequence.
# Unique ID: protein_gi + phosphorylation site in protein sequence.
# system.time({
#   summary_df_of_unique_proteins_with_sites = get_summary_with_unique_sites(
#     combined_df_with_mapped_gene_symbol, species = 'human', fasta_type = 'refseq'
#   )
# })
head(summary_df_of_unique_proteins_with_sites)
```

##	AA_in_protein	AA_in_peptide	Sequence		
## gi 39930517_s161	s161	s11	AAAAAATAPPsPGPAQPGPR		
## gi 4758248_s281	s281	s5	AAALsLSTLASPK		
## gi 4758248_s283	s283	s7	AAALSLsTLASPK		
## gi 4758248_s287	s287	s11	AAALSLSTLASPK		
## gi 4758248_s292	s292	s3	GGsGTAGTEPSDIIIPLR		
## gi 4758248_t284	t284	t8	AAALSLSTLASPK		
##	ID	Modification	GeneSymbol	Exp027012	
## gi 39930517_s161	gi 39930517	Phospho (ST)(11)	SAMD1	191101700	
## gi 4758248_s281	gi 4758248	Phospho (ST)(5)	EFNB1	225996500	
## gi 4758248_s283	gi 4758248	Phospho (ST)(7)	EFNB1	25986220	
## gi 4758248_s287	gi 4758248	Phospho (ST)(11)	EFNB1	19612020	
## gi 4758248_s292	gi 4758248	Phospho (ST)(3)	EFNB1	0	
## gi 4758248_t284	gi 4758248	Phospho (ST)(8)	EFNB1	0	
##	Exp027013	Exp027014	Exp027015	Exp027016	Exp027017
## gi 39930517_s161	72397330	0	32192620	231611600	124170700
## gi 4758248_s281	148918100	77889940	360054600	226245100	189362500
## gi 4758248_s283	12633960	0	0	0	0
## gi 4758248_s287	17807950	19357340	79808370	84246970	56323030
## gi 4758248_s292	0	0	0	0	0
## gi 4758248_t284	0	12164570	0	34897000	0
##	Exp027018	Exp027019	Exp027020	Exp027021	Exp027022
## gi 39930517_s161	28650550	31780240	123238200	30951010	0
## gi 4758248_s281	203163300	159109000	112891900	117936700	130344300
## gi 4758248_s283	0	0	0	0	0
## gi 4758248_s287	48652040	33331610	0	26782070	59331840
## gi 4758248_s292	0	0	0	0	0
## gi 4758248_t284	0	0	0	0	0
##	Exp027023	Exp027024	Exp027025	Exp027026	Exp027027
## gi 39930517_s161	159176400	0	0	0	36968750
## gi 4758248_s281	136386600	40931780	40848750	84373550	18245820
## gi 4758248_s283	0	0	0	0	0
## gi 4758248_s287	81300420	11794550	25698570	48053290	11260400
## gi 4758248_s292	0	0	0	0	0
## gi 4758248_t284	4121336	0	0	53214350	0
##	Exp027028	Exp027029	Exp027030	Exp027031	Exp027032
## gi 39930517_s161	345538800	167434500	232864900	0	341993300
## gi 4758248_s281	449307600	27093610	205768900	140357500	203682300
## gi 4758248_s283	0	0	7733698	0	5456458
## gi 4758248_s287	49733950	17026620	33251190	68674110	82809850
## gi 4758248_s292	0	0	0	0	0
## gi 4758248_t284	12024320	0	0	0	0
##	Exp027033	Exp027034	Exp027035	Exp027036	Exp027037
## gi 39930517_s161	171803500	0	38057890	109961100	167525000
## gi 4758248_s281	71224800	170784100	153310800	41458280	70873710
## gi 4758248_s283	0	0	0	0	0
## gi 4758248_s287	20488140	66844210	94660320	26547410	0
## gi 4758248_s292	0	0	0	0	0
## gi 4758248_t284	0	0	0	6888869	0
##	Exp027038	Exp027039	Exp027040	Exp027041	Exp027042
## gi 39930517_s161	165389700	51471910	143915200	356015900	208103900
## gi 4758248_s281	99639770	132810800	78469830	92153170	344426900
## gi 4758248_s283	0	9818745	0	0	0
## gi 4758248_s287	0	44516010	107605300	33192480	0
## gi 4758248_s292	0	0	7382834	0	0
## gi 4758248_t284	0	40079320	0	35790700	21361960
##	Exp027043	Exp027044	Exp027045	Exp027046	Exp027047
## gi 39930517_s161	101188100	0	54273770	63307160	83332280
## gi 4758248_s281	134677700	68013420	27048770	52237850	65281570
## gi 4758248_s283	0	0	0	0	0
## gi 4758248_s287	0	0	0	0	0
## gi 4758248_s292	0	0	0	0	0
## gi 4758248_t284	8331117	0	0	0	0
##	Exp027048	Exp027049	Exp027050		
## gi 39930517_s161	162916400	120589000	271443600		
## gi 4758248_s281	116332000	0	0		

```
## gi|4758248_s283      0      0      0
## gi|4758248_s287    22553820      0      0
## gi|4758248_s292      0      0      0
## gi|4758248_t284      0      0      0
```

Data normalization

PhosMap provides two kinds of normalizations. 1. PhosMap allowed for a total sum scaling normalization.

```
# Imputation with the next order of magnitude of the minimum except for zero.
# Filtering data only including phosphorylation site.
phospho_data_filtering_STY_and_normalization_list <- get_normalized_data_of_psites(
  summary_df_of_unique_proteins_with_sites,
  phosphorylation_exp_design_info_file_path,
  topN = NA, mod_types = c('S', 'T', 'Y')
)
phospho_data_filtering_STY <- phospho_data_filtering_STY_and_normalization_list$ptypes_area_df_with_id
phospho_data_filtering_STY_and_normalization <-
phospho_data_filtering_STY_and_normalization_list$ptypes_fot5_df_with_id
head(phospho_data_filtering_STY_and_normalization)
```

2. If having matched proteomics data with phosphoproteomics, PhosMap allowed for normalizing phosphoproteomics data based on proteomics data.

```
# Based on phospho_data_filtering_STY_and_normalization
ID <- paste(phospho_data_filtering_STY_and_normalization$GeneSymbol,
  phospho_data_filtering_STY_and_normalization$AA_in_protein,
  sep = '_')
Value <- phospho_data_filtering_STY_and_normalization[, -seq(1,6)]
phospho_data <- data.frame(ID, Value)
phospho_data_rownames <- paste(phospho_data_filtering_STY_and_normalization$GI,
  phospho_data_filtering_STY_and_normalization$GeneSymbol,
  phospho_data_filtering_STY_and_normalization$AA_in_protein,
  sep = '_')
rownames(phospho_data) <- phospho_data_rownames
# Further normalize phosphoproteomics data based on proteomics data
# The configurations of function see help document.
data_frame_normalization_with_control_no_pair <- normalize_phos_data_to_profiling(
  phospho_data, profiling_data_normalized,
  phosphorylation_exp_design_info_file_path,
  profiling_exp_design_info_file_path,
  control_label = '0',
  pair_flag = FALSE
)
head(data_frame_normalization_with_control_no_pair)
```

Data analysis

PhosMap incorporated four analysis modules, including clustering and differential expression analysis, time course analysis, kinase-substrate enrichment analysis to find activated/deactivated kinases and motif enrichment analysis.

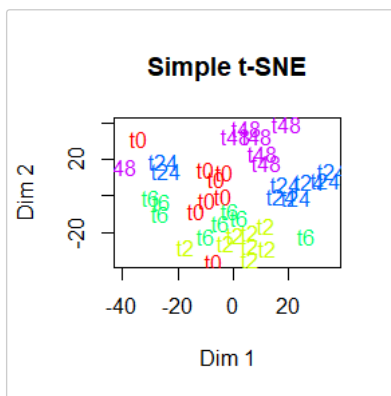
Clustering and differential expression analysis

- In PhosMap, Clustering methods allowed for t-SNE and PCA
 - t-SNE

```
# Clustering: t-SNE or PCA
expr_data_frame <- data_frame_normalization_with_control_no_pair
# t-SNE using all experiments
visualization_with_simple_tsne(expr_data_frame, group)

## Read the 39 x 39 data matrix successfully!
## Using no_dims = 2, perplexity = 10.000000, and theta = 0.500000
## Computing input similarities...
## Normalizing input...
## Building tree...
## - point 0 of 39
## Done in 0.00 seconds (sparsity = 0.902038)!
## Learning embedding...
## Iteration 50: error is 58.888641 (50 iterations in 0.01 seconds)
## Iteration 100: error is 56.034252 (50 iterations in 0.01 seconds)
## Iteration 150: error is 60.450825 (50 iterations in 0.01 seconds)
## Iteration 200: error is 56.571225 (50 iterations in 0.01 seconds)
## Iteration 250: error is 58.820953 (50 iterations in 0.01 seconds)
## Iteration 300: error is 1.697285 (50 iterations in 0.01 seconds)
## Iteration 350: error is 1.398376 (50 iterations in 0.01 seconds)
## Iteration 400: error is 0.920080 (50 iterations in 0.01 seconds)
## Iteration 450: error is 0.650832 (50 iterations in 0.00 seconds)
```

```
## Iteration 500: error is 0.494792 (50 iterations in 0.00 seconds)
## Fitting performed in 0.06 seconds.
```



- PCA

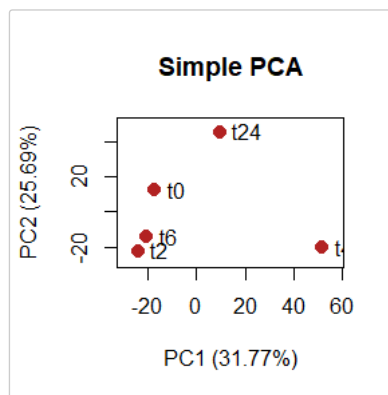
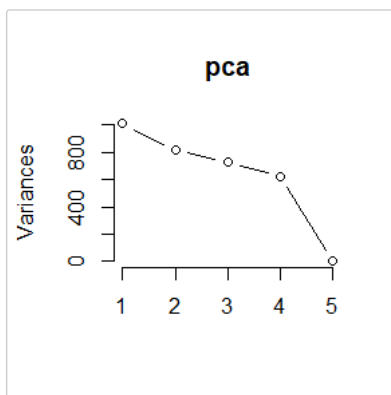
```
expr_ID <- as.vector(expr_data_frame[,1])
expr_Valule <- expr_data_frame[, -1]
expr_Valule_mean <- NULL
expr_Valule_row <- nrow(expr_Valule)
for(i in 1:expr_Valule_row){
  x <- as.vector(unlist(expr_Valule[i,]))
  x_m <- tapply(x, group, mean)
  expr_Valule_mean <- rbind(expr_Valule_mean, x_m)
}
group_levels = levels(group)
colnames(expr_Valule_mean) <- group_levels
expr_df <- data.frame(expr_ID, expr_Valule_mean)
```

```
## Warning in data.row.names(row.names, rowsi, i): some row.names duplicated:
```

```
##
```

```
2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,
```

```
# PCA using mean value in group for comparison with original literature
visualization_with_simple_pca(expr_df)
```



- In PhosMap, differential expression analysis methods allowed for limma, SAM and ANOVA Data preparation (t2 VS t0)

```
# Differently expressed Proteins/Genes analysis
# t2 vs t0
expr_data_frame <- data_frame_normalization_with_control_no_pair[,1:17] # phosphoproteomics data
normalized by proteomics data
# select phosphorylation sites with greater variation
expr_data_frame_var <- apply(expr_data_frame, 1, function(x){
  var(x[-1])
})
index_of_kept <- which(expr_data_frame_var>1)
expr_data_frame <- expr_data_frame[index_of_kept,]

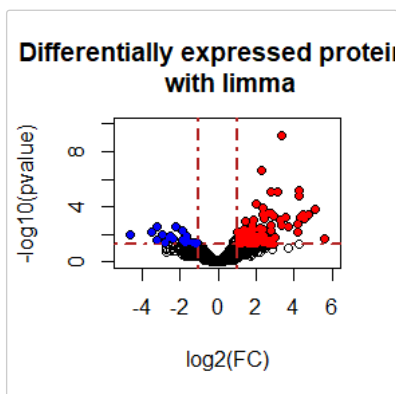
# group information (t0 vs t2)
deps_group_levels <- c('t0', 't2')
deps_group <- factor(as.vector(group)[1:16], levels = deps_group_levels)
```

```
- limma
```

```
# (1) Limma
limma_results_df <- analysis_deps_limma(expr_data_frame, deps_group, deps_group_levels, log2_label =
FALSE, adjust_method = 'none')
```

```
##
## The matrix of experiment design.  group0 group2
## 1      1      0
## 2      1      0
## 3      1      0
## 4      1      0
## 5      1      0
## 6      1      0
## 7      1      0
## 8      1      0
## 9      0      1
## 10     0      1
## 11     0      1
## 12     0      1
## 13     0      1
## 14     0      1
## 15     0      1
## 16     0      1
## attr("assign")
## [1] 1 1
## attr("contrasts")
## attr("contrasts")$group
## [1] "contr.treatment"
##
##
## The combination of pairwise comparison(s).
## t2-t0
##
## The matrix of comparison statement, compare other groups with control.      Contrasts
## Levels t2-t0
##      t0      -1
##      t2       1

limma_results_df$ID <- apply(limma_results_df, 1, function(x){
  x = strsplit(x, '_')[[1]]
  paste(x[2], x[3], sep = '_')
})
visualization_deps_with_scatter(limma_results_df, minFC = 2, minPvalue = 0.05, main = 'Differentially
expressed proteins \n with limma',
                               show_text = FALSE, min_up_text = 70, min_down_text = 70)
```



- SAM

```
# (2) SAM
sam_results_list <- analysis_deps_sam(expr_data_frame, deps_group, log2_label = FALSE, nperms = 100,
rand = NULL, minFDR = 0.05, samr_plot = TRUE)
sam_results <- rbind(sam_results_list$genes_up_df, sam_results_list$genes_down_df)
```

- ANOVA

```
# (3) anova
anova_result_df <- analysis_deps_anova(expr_data_frame, deps_group, log2_label = FALSE, return_padjust
= TRUE, adjust_method = 'BH')
visualization_deps_with_scatter(anova_result_df, minFC = 2, minPvalue = 0.05, main = 'Differentially
expressed proteins \n with anova',
                               show_text = FALSE, min_up_text = 15, min_down_text = 15)
```

Time course analysis

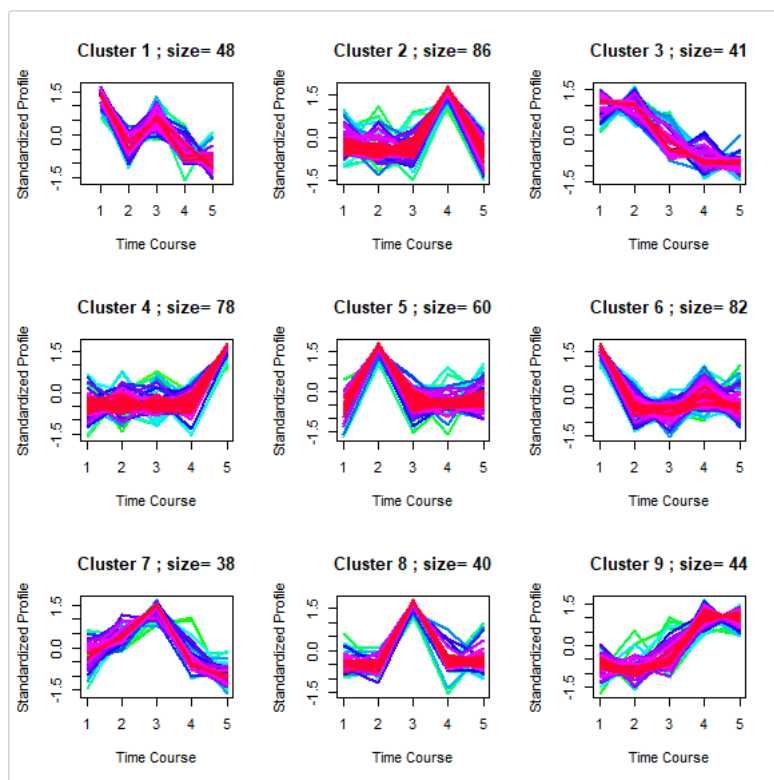
Fuzzy clustering was applied to time course analysis for discovering patterns associated with time points in PhosMap. The corresponding line chart combined with membership for each cluster was also drawn.

```
group_levels <- levels(group)
# fuzzy c-means clustering
```

```

set.seed(1000)
fuzzy_clustObj <- visualization_fuzzycluster(
  fuzzy_input_df, group, group_levels,
  k_cluster=9, iteration = 100,
  mfrow = c(3,3), min_mem = 0.1,
  plot = TRUE
)

```



Time course analysis example

```

# clusters information
clusterS_info <- fuzzy_clustObj$cluster
clusterS_names <- names(clusterS_info)
clusters_df <- data.frame(clusterS_names, clusterS_info)
# write.csv(clusters_df, 'clusters_df.csv', row.names = TRUE)

```

Kinase activity prediction to find activated/deactivated kinases

In PhosMap, three kinase activity prediction methods were included: KSEA, multiple linear regression (MLR) and Mean Value. - Data preparation

```

# For early and late response
# early -> clusterS_info==1
# late -> clusterS_info==2
cluster_flag <- 'early'
cluster_symbol <- clusterS_names[clusterS_info==1]
expr_data_frame <- data_frame_normalization_with_control_no_pair
index_of_cluster <- match(cluster_symbol, expr_data_frame$ID)
cluster_df <- expr_data_frame[index_of_cluster,]

# KSEA

# Perform KSEA
summary_df_list_from_ksea_cluster <- get_summary_from_ksea(cluster_df, species = 'human', log2_label =
FALSE, ratio_cutoff = 3)

##
## Starting KSEA
## completing: 1 / 39
## completed: 1 / 39
## completing: 2 / 39
## completed: 2 / 39
## completing: 3 / 39
## completed: 3 / 39
## completing: 4 / 39
## completed: 4 / 39
## completing: 5 / 39
## completed: 5 / 39
## completing: 6 / 39

```

```

## completed: 6 / 39
## completing: 7 / 39
## completed: 7 / 39
## completing: 8 / 39
## completed: 8 / 39
## completing: 9 / 39
## completed: 9 / 39
## completing: 10 / 39
## completed: 10 / 39
## completing: 11 / 39
## completed: 11 / 39
## completing: 12 / 39
## completed: 12 / 39
## completing: 13 / 39
## completed: 13 / 39
## completing: 14 / 39
## completed: 14 / 39
## completing: 15 / 39
## completed: 15 / 39
## completing: 16 / 39
## completed: 16 / 39
## completing: 17 / 39
## completed: 17 / 39
## completing: 18 / 39
## completed: 18 / 39
## completing: 19 / 39
## completed: 19 / 39
## completing: 20 / 39
## completed: 20 / 39
## completing: 21 / 39
## completed: 21 / 39
## completing: 22 / 39
## completed: 22 / 39
## completing: 23 / 39
## completed: 23 / 39
## completing: 24 / 39
## completed: 24 / 39
## completing: 25 / 39
## completed: 25 / 39
## completing: 26 / 39
## completed: 26 / 39
## completing: 27 / 39
## completed: 27 / 39
## completing: 28 / 39
## completed: 28 / 39
## completing: 29 / 39
## completed: 29 / 39
## completing: 30 / 39
## completed: 30 / 39
## completing: 31 / 39
## completed: 31 / 39
## completing: 32 / 39
## completed: 32 / 39
## completing: 33 / 39
## completed: 33 / 39
## completing: 34 / 39
## completed: 34 / 39
## completing: 35 / 39
## completed: 35 / 39
## completing: 36 / 39
## completed: 36 / 39
## completing: 37 / 39
## completed: 37 / 39
## completing: 38 / 39
## completed: 38 / 39
## completing: 39 / 39
## completed: 39 / 39
## Ending KSEA
## Extracting information data frame derived from KSEA
## ***** Regulation direction from KSEA *****
## ***** Pvalue from KSEA *****
## ***** Activity from KSEA *****
## ***** Kinase_site_substrate quantification matrix after KSEA *****
##
## KSEA OK! ^_^

# Activity of regulons for regulation
ksea_regulons_activity_df_cluster <- summary_df_list_from_ksea_cluster$ksea_regulons_activity_df
ksea_id_cluster <- as.vector(ksea_regulons_activity_df_cluster[,1])
ksea_value_cluster <- ksea_regulons_activity_df_cluster[,-1]
if(FALSE){
  # Pvalue of regulons for regulation
  ksea_regulons_pvalue_cluster <- summary_df_list_from_ksea_cluster$ksea_regulons_pvalue_df
  # Activity of regulons for regulation
  ksea_regulons_activity_cluster <- summary_df_list_from_ksea_cluster$ksea_regulons_activity_df
  # Expression ratio of regulons for regulation

```



```
## Warning: package 'pheatmap' was built under R version 3.3.3
```



KSEA method

◦ MLR

```
# get kinase activity matrix with multiple linear regression (mlr) method
kinase_activity_df_mlr <- get_ka_by_mean_or_mlr(cluster_df, species = 'human', log2_label = TRUE,
method = 'mlr')
```

◦ Mean value

```
# get kinase activity matrix with mean value method
kinase_activity_df_mean <- get_ka_by_mean_or_mlr(cluster_df, species = 'human', log2_label = TRUE,
method = 'mean')
```

Motif enrichment analysis (MEA)

PhosMap allowed for performing MEA on user defined phosphopeptides lists. - Data preparation

```
# *** foreground ***
foreground_data <- phospho_data_filtering_STY_and_normalization # pre-processed data
foreground_sequence <- as.vector(foreground_data$Sequence)
GI <- as.vector(foreground_data$GI)
Sequence <- as.vector(foreground_data$Sequence)
AA_in_protein <- as.vector(foreground_data$AA_in_protein)

# *** required parameters ***
fixed_length <- 15
species <- 'human'
motifx_pvalue <- 0.01

# get foreground data frame
# foreground_df = get_aligned_seq_for_mea(ID, Sequence, AA_in_protein, fixed_length, species = 'human',
# fasta_type = 'refseq')

# get background data frame
# background_df = get_global_background_df(species = 'human', fasta_type = 'refseq')
```

◦ Motif enrichment analysis

```
# construct foreground and background
# To facilitate testing the module, select an appropriate number of items at random.
foreground <- as.vector(foreground_df$aligned_seq)
foreground <- foreground[sample(length(foreground), 1000)]
background <- as.vector(background_df$Aligned_Seq)
background <- background[sample(length(background), 10000)]

motifs_list <- mea_based_on_background(foreground, AA_in_protein, background, motifx_pvalue)

## Start executing motifx and find motif pattern.
## Foreground sequences: 1000.
## Background sequences: 10000.
## Phosphorylation: [STY] exists in foreground.
## Motifx pvalue cutoff: 0.01.
## Motifx analysis OK! ^_^
## $S
##          motif      score foreground_matches foreground_size
## 1  ..S....SP.K.... 48.000000           9           818
## 2  .....SP...M. 32.000000           4           809
## 3  .A....SP.K.... 42.225166           7           805
## 4  .....SSP...K. 42.148780          10           798
## 5  .K...S.SP..... 40.251228           6           788
## 6  ....D.SP...K. 37.572669           6           782
## 7  .....SP...K.T 37.684140           6           776
## 8  ....P.SPT.... 36.595410           9           770
## 9  ...R..SP..L... 36.451484           8           761
## 10 ...V..SP....K 36.322978           7           753
## 11 KK....SP..... 36.185583           3           746
## 12 ...P..SPK.... 35.736980           6           743
## 13 ...D...SPL.... 35.599662           5           737
## 14 E.....SP.K.... 35.670512           5           732
## 15 .A....SP....T 35.339136           4           727
## 16 ...P...SP.T... 35.264427           4           723
## 17 ...R.A.SP..... 34.886288           5           719
## 18 ..N....SP...A. 34.691114           3           714
## 19 .A.K...SP..... 34.752471           5           711
## 20 ...AP.SP..... 34.757912           6           706
## 21 .KG...SP..... 34.505420           3           700
## 22 ..L....SP...K. 34.269830           4           697
## 23 .....DSPT.... 34.263249           4           693
## 24 .H....SP...P. 34.302185           4           689
## 25 ...K..SPP.... 34.119689           4           685
```

## 26	..L....SPK....	34.119283	5	681
## 27SP..H.P.	34.149745	3	676
## 28	..Q....SP...R..	34.177619	4	673
## 29	.G....SPP....	34.220087	4	669
## 30	..D.L.SP.....	34.271524	3	665
## 31	.P....SP....Y.	34.170207	3	662
## 32S.SP.T....	34.205550	4	659
## 33	..N....SP.....K	34.010728	3	655
## 34SP.....	16.000000	184	652
## 35R..S..W....	32.000000	1	468
## 36	.PR.R..S.....	38.157168	4	467
## 37	...RRR.S.....	37.693105	5	463
## 38	...RS.S..N....	38.073104	4	458
## 39	...QRS.S.....	36.625163	3	454
## 40	..L.RT.S.....	36.602283	4	451
## 41R..S...DE..	36.035171	3	447
## 42	...RS.S..P....	36.179842	3	444
## 43	.LR.R..S.....	35.559971	3	441
## 44SD.E...D	47.954590	8	438
## 45	.R..RR.S.....	35.294127	3	430
## 46	S...R..S.D....	35.381203	6	427
## 47	.D....S.DE....	43.817742	4	421
## 48	...R..S.E...H	35.348966	3	417
## 49ESD.E....	42.458524	4	414
## 50	S..KR..S.....	34.988344	3	410
## 51	AG..R..S.....	34.545411	4	407
## 52AS.DE....	41.654105	4	403
## 53	.R..R..S....S.	34.719963	3	399
## 54	...R..ST..S...	34.660085	4	396
## 55	.L..R..S.....P	34.380718	3	392
## 56S..EDD..	38.843821	5	389
## 57	...R..S..S.A..	33.106304	3	384
## 58	PR..R..S.....	31.503238	2	381
## 59ESE.E....	36.661413	7	379
## 60	...R..SL.L....	31.074906	4	372
## 61	L.....SD.E....	33.614273	3	368
## 62	R...R..SF.....	29.281407	2	365
## 63	T.....SD.E....	30.759346	3	363
## 64	.H..R..S..S....	28.298977	2	360
## 65S.S...W..	25.849456	2	358
## 66	.S..RR.S.....	27.608165	3	356
## 67	.P..E..SD.....	28.606337	3	353
## 68	...KS.S.E....	31.095118	3	350
## 69	..N..S.S.....P	28.079446	3	347
## 70	..L.R..S..G....	26.574005	3	344
## 71	..G...S..E.D..	28.800424	3	341
## 72ES.DG....	27.244695	2	338
## 73	...RN.S....E.	25.679071	2	336
## 74	F.R..S.S.....	27.096369	1	334
## 75	.S.....SE.E....	27.429643	3	333
## 76	...KS.S.G....	26.956968	2	330
## 77	R.....S.DE....	26.774816	2	328
## 78	W.....S.E....	23.166292	1	326
## 79	...R..S..S...Q	25.153312	2	325
## 80	..K....S.E...E	26.818129	3	323
## 81	.A...S.S.D....	25.456919	1	320
## 82	..E.GS.S.....	25.035183	3	319
## 83	TH..R..S.....	24.652101	2	316
## 84	...S..S..E...R	25.871468	3	314
## 85	...S..S.E..E..	25.645588	3	311
## 86	...F..S.D.D...	25.115521	2	308
## 87	...R..S.....	6.181342	41	306
## 88S.DE..S.	24.413199	2	265
## 89	RN...R.S.....	25.500564	2	263
## 90	.G.R..SS.....	25.536039	3	261
## 91	R.....S...TA.	24.151322	2	258
## 92	...A..SD.E....	23.698785	3	256
## 93	.R.S.R.S.....	23.737604	3	253
## 94	..L....S.ED....	23.469117	3	250
## 95SLDG....	23.017292	2	247
## 96	.K.S...S..S....	23.367358	2	245
## 97	..HR..SS.....	23.110557	2	243
## 98	.R.KS..S.....	26.152294	3	241
## 99	..R..SGS.....	23.684658	4	238
## 100	..RE...S.E....	22.405664	2	234
## 101	W.....SE.....	20.117637	1	232
## 102	...R..SS....G.	22.748544	4	231
## 103	..VRS..S.....	23.255797	2	227
## 104SE...EE.	24.244164	3	225
## 105	..E..S.S...I..	22.189385	2	222
## 106SSPD....	28.099112	3	220
## 107	.A..D..S..S....	22.104210	2	217
## 108	.R..S..S.P....	23.183744	2	215
## 109	...K..S...SV.	22.847138	4	213
## 110	..K.S..S...I..	21.875225	2	209
## 111	.Q....SD..A...	21.946674	2	207
## 112	.K..L..S.E....	21.958745	2	205

## 113S.SE.D....	21.832711	3	203
## 114	.A.....S..S..Q.	21.667962	2	200
## 115	.R..SN.S.....	21.550853	1	198
## 116	R.RS...S.....	22.123170	2	197
## 117	...S..S.....	3.130885	39	195
## 118SSPP....	23.840535	3	156
## 119	..T.G..S....R.	22.071777	2	153
## 120	A...K..S....T..	22.522901	2	151
## 121	..PS...S.....E	21.920473	3	149
## 122	D..R...S..S....	21.872994	2	146
## 123DSE....E.	22.161527	2	144
## 124S.D.....	2.919707	16	142
## 125	R.....S..S..M.	21.072067	2	126
## 126	...GG.S.P.....	20.929136	2	124
## 127S.E.....	2.974737	19	122
## 128	..ESQ..S.....	20.865939	2	103
## 129S..S....	2.695456	20	101
## 130	.K.....SFS.....	21.259819	3	81
## 131	...G..S.....	2.686796	14	78
## 132	M...K..SE.....	21.092487	1	64
## 133	K.....SSV.....	20.884506	2	63
## 134	P.....SE...A..	21.421576	2	61
## 135	.L..K..S.....G	20.833586	2	59
## 136S.P..A.G	20.694579	2	57
## 137	E.E....S....K..	20.647539	2	55
## 138	WT....S.....	18.200707	1	53
## 139	...K..S.....	2.249685	10	52
## 140S.....S	2.026490	9	42
##	background_matches	background_size	fold_increase	
## 1	0	5005	Inf	
## 2	0	5005	Inf	
## 3	0	5005	Inf	
## 4	0	5005	Inf	
## 5	0	5005	Inf	
## 6	0	5005	Inf	
## 7	0	5005	Inf	
## 8	0	5005	Inf	
## 9	0	5005	Inf	
## 10	0	5005	Inf	
## 11	0	5005	Inf	
## 12	0	5005	Inf	
## 13	0	5005	Inf	
## 14	0	5005	Inf	
## 15	0	5005	Inf	
## 16	0	5005	Inf	
## 17	0	5005	Inf	
## 18	0	5005	Inf	
## 19	0	5005	Inf	
## 20	0	5005	Inf	
## 21	0	5005	Inf	
## 22	0	5005	Inf	
## 23	0	5005	Inf	
## 24	0	5005	Inf	
## 25	0	5005	Inf	
## 26	0	5005	Inf	
## 27	0	5005	Inf	
## 28	0	5005	Inf	
## 29	0	5005	Inf	
## 30	0	5005	Inf	
## 31	0	5005	Inf	
## 32	0	5005	Inf	
## 33	0	5005	Inf	
## 34	352	5005	4.012653	
## 35	0	4653	Inf	
## 36	0	4653	Inf	
## 37	0	4653	Inf	
## 38	0	4653	Inf	
## 39	0	4653	Inf	
## 40	0	4653	Inf	
## 41	0	4653	Inf	
## 42	0	4653	Inf	
## 43	0	4653	Inf	
## 44	0	4653	Inf	
## 45	0	4653	Inf	
## 46	0	4653	Inf	
## 47	0	4653	Inf	
## 48	0	4653	Inf	
## 49	0	4653	Inf	
## 50	0	4653	Inf	
## 51	0	4653	Inf	
## 52	0	4653	Inf	
## 53	0	4653	Inf	
## 54	0	4653	Inf	
## 55	0	4653	Inf	
## 56	0	4653	Inf	
## 57	0	4653	Inf	
## 58	0	4653	Inf	

## 59	0	4653	Inf
## 60	0	4653	Inf
## 61	0	4653	Inf
## 62	0	4653	Inf
## 63	0	4653	Inf
## 64	0	4653	Inf
## 65	0	4653	Inf
## 66	0	4653	Inf
## 67	0	4653	Inf
## 68	0	4653	Inf
## 69	0	4653	Inf
## 70	0	4653	Inf
## 71	0	4653	Inf
## 72	0	4653	Inf
## 73	0	4653	Inf
## 74	0	4653	Inf
## 75	0	4653	Inf
## 76	0	4653	Inf
## 77	0	4653	Inf
## 78	0	4653	Inf
## 79	0	4653	Inf
## 80	0	4653	Inf
## 81	0	4653	Inf
## 82	0	4653	Inf
## 83	0	4653	Inf
## 84	0	4653	Inf
## 85	0	4653	Inf
## 86	0	4653	Inf
## 87	269	4653	2.317625
## 88	0	4384	Inf
## 89	0	4384	Inf
## 90	0	4384	Inf
## 91	0	4384	Inf
## 92	0	4384	Inf
## 93	0	4384	Inf
## 94	0	4384	Inf
## 95	0	4384	Inf
## 96	0	4384	Inf
## 97	0	4384	Inf
## 98	0	4384	Inf
## 99	0	4384	Inf
## 100	0	4384	Inf
## 101	0	4384	Inf
## 102	0	4384	Inf
## 103	0	4384	Inf
## 104	0	4384	Inf
## 105	0	4384	Inf
## 106	0	4384	Inf
## 107	0	4384	Inf
## 108	0	4384	Inf
## 109	0	4384	Inf
## 110	0	4384	Inf
## 111	0	4384	Inf
## 112	0	4384	Inf
## 113	0	4384	Inf
## 114	0	4384	Inf
## 115	0	4384	Inf
## 116	0	4384	Inf
## 117	517	4384	1.695938
## 118	0	3867	Inf
## 119	0	3867	Inf
## 120	0	3867	Inf
## 121	0	3867	Inf
## 122	0	3867	Inf
## 123	0	3867	Inf
## 124	182	3867	2.394057
## 125	0	3685	Inf
## 126	0	3685	Inf
## 127	261	3685	2.198825
## 128	0	3424	Inf
## 129	336	3424	2.017916
## 130	0	3088	Inf
## 131	233	3088	2.378783
## 132	0	2855	Inf
## 133	0	2855	Inf
## 134	0	2855	Inf
## 135	0	2855	Inf
## 136	0	2855	Inf
## 137	0	2855	Inf
## 138	0	2855	Inf
## 139	218	2855	2.518525
## 140	227	2637	2.489301
##			
## \$T			
##	motif	score	foreground_matches foreground_size
## 1TP...M..	32.000000	1 179
## 2	...T..TP.K....	36.858761	4 178

## 3A.TP...S.	36.347203	4	174
## 4	...S...TP....K	35.767435	3	170
## 5	.E.K...TP.....	35.113820	2	167
## 6	...G.QTP.....	35.023592	3	165
## 7TPT.S...	34.782740	3	162
## 8N.TPP.....	34.660568	3	159
## 9S.TP...E.	34.763790	4	156
## 10	.T....TP....L.	34.615743	3	152
## 11TPKK....	32.545203	3	149
## 12	.P.K...TP.....	30.417277	2	146
## 13	...D..KTP.....	28.996989	2	144
## 14TP.....	9.811600	33	142
## 15	...YSPT.....	27.256459	3	109
## 16	DE....TD.....	25.284548	2	106
## 17	..E...S.TD.....	23.824752	2	104
## 18	...R..ST.....P	26.738103	3	102
## 19	RR..S..T.....	23.578118	2	99
## 20SPTK.....	22.663644	2	97
## 21	.V....TD....E	23.255949	2	95
## 22	..SP...T...S..	21.717993	3	93
## 23	R...RS..T.....	22.087279	2	90
## 24	..QR..ST.....	22.926109	2	88
## 25S.T..E.P..	21.613665	3	86
## 26	K..K..ET.....	20.913735	2	83
## 27	..K...ST..L...	21.078795	2	81
## 28	.S.....TDE....	21.294180	2	79
## 29	..Q.S..T...S..	20.741222	2	77
## 30	P....S.T....E.	20.451707	1	75
## 31	..P...DT.E....	21.476484	2	74
## 32T.D...W	18.421110	1	72
## 33S.T.....	2.219047	13	71
## 34T...S..	2.838347	12	58
## 35	KK.....TI.....	21.310701	2	46
## 36	..E...DT.....E	21.197757	2	44
## 37	G....KAT.....	20.524600	2	42

background_matches background_size fold_increase

## 1	0	3265	Inf
## 2	0	3265	Inf
## 3	0	3265	Inf
## 4	0	3265	Inf
## 5	0	3265	Inf
## 6	0	3265	Inf
## 7	0	3265	Inf
## 8	0	3265	Inf
## 9	0	3265	Inf
## 10	0	3265	Inf
## 11	0	3265	Inf
## 12	0	3265	Inf
## 13	0	3265	Inf
## 14	214	3265	3.545643
## 15	0	3051	Inf
## 16	0	3051	Inf
## 17	0	3051	Inf
## 18	0	3051	Inf
## 19	0	3051	Inf
## 20	0	3051	Inf
## 21	0	3051	Inf
## 22	0	3051	Inf
## 23	0	3051	Inf
## 24	0	3051	Inf
## 25	0	3051	Inf
## 26	0	3051	Inf
## 27	0	3051	Inf
## 28	0	3051	Inf
## 29	0	3051	Inf
## 30	0	3051	Inf
## 31	0	3051	Inf
## 32	0	3051	Inf
## 33	257	3051	2.173672
## 34	215	2794	2.688693
## 35	0	2579	Inf
## 36	0	2579	Inf
## 37	0	2579	Inf

\$Y

motif score foreground_matches foreground_size

1 S....G.Y.G..... 20.58452 2 3

background_matches background_size fold_increase

1 0 1688 Inf

Find sequences in foreground that are mapped to specific motif
foreground_sequences_mapped_to_motifs <- get_foreground_seq_to_motifs(motifs_list, foreground)

Find sequences in foreground that are mapped to specific motif.

```
# Find data frame in foreground that are mapped to specific motif
foreground_df_mapped_to_motifs <- get_foreground_df_to_motifs(foreground_sequences_mapped_to_motifs,
foreground, foreground_df)

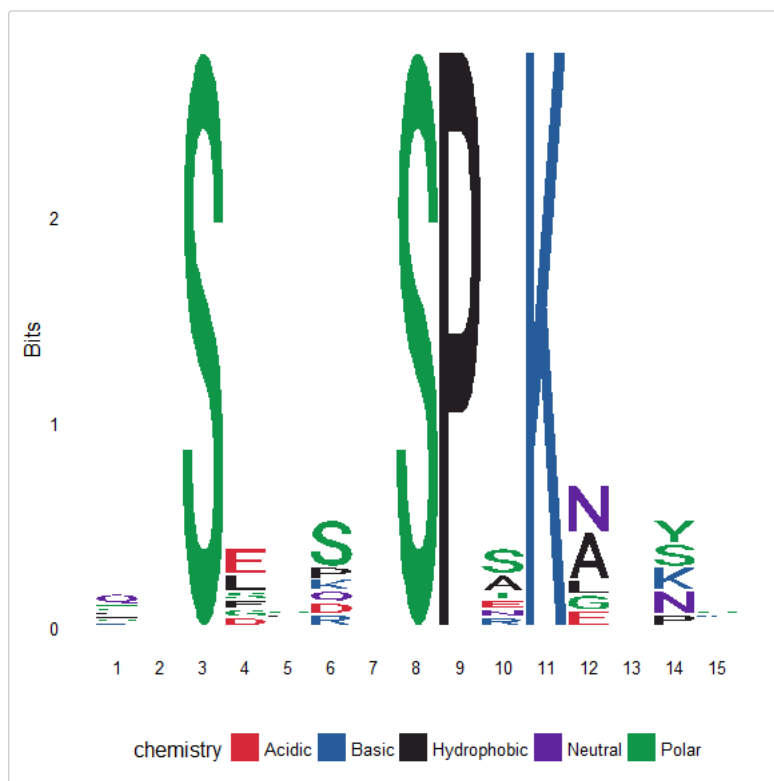
##
## Find data frame in foreground that are mapped to specific motif.

  ◦ Plot motif logo

# The data can be used for plotting logo of specific motif: foreground_sequences_mapped_to_motifs
# plotting logo: Q.....S.....
require(ggseqlogo)

## Loading required package: ggseqlogo

ggseqlogo(foreground_sequences_mapped_to_motifs[[1]])
```



Plot motif logo

```
if(TRUE){
  # batch plot and count peptides for each motif
  foreground_sequences_mapped_to_motifs_count <- length(foreground_sequences_mapped_to_motifs)
  motifs <- names(foreground_sequences_mapped_to_motifs)
  peptides_count <- NULL
  for(i in seq_len(foreground_sequences_mapped_to_motifs_count)){
    l_i <- foreground_sequences_mapped_to_motifs[[i]]
    peptides_count <- c(peptides_count, length(l_i))
  }
  motifs_peptides_count_df <- data.frame(motifs, peptides_count)
  # quantile(peptides_count, seq(0,1,0.05))
  if(FALSE){
    plot_seqlogo(BASE_DIR, foreground_sequences_mapped_to_motifs, plot_min_seqs = 25)
  }
}
```

- Assign quantitative values of peptides to their motif

```
# Select motifs at least having 50 peptides
# Assign quantitative values of peptides to their motif
foreground_value <- foreground_data[, -c(seq(1,6))]
min_seqs <- 50
index_of_motifs <- which(peptides_count >= min_seqs)
motif_group_m_ratio_df <- NULL
for(i in index_of_motifs){
  motif <- motifs[i]
  aligned_peptides <- foreground_sequences_mapped_to_motifs[[i]]
```

```

index_of_match <- match(aligned_peptides, foreground_df$aligned_seq)
motif_value <- foreground_value[index_of_match,]
motif_value_colsum <- colSums(motif_value)
motif_group_m <- tapply(motif_value_colsum, group, mean)
motif_group_m_ratio <- motif_group_m/motif_group_m[1]
motif_group_m_ratio_df <- rbind(motif_group_m_ratio_df, motif_group_m_ratio)
}

motifs_subset <- motifs[index_of_motifs]
peptides_count_subset <- peptides_count[index_of_motifs]
rownames(motif_group_m_ratio_df) <- paste(motifs_subset, peptides_count_subset)

# The matrix is import inot pheatmap
motif_group_m_ratio_df_mat <- as.matrix(motif_group_m_ratio_df)

# plot pheatmap
if(TRUE){
  # library(pheatmap)
  # breaks and colors setting
  breaks_1 <- seq(0, 0.5, 0.1)
  colors_1 <- colorRampPalette(c('green', 'blue'))(length(breaks_1)-1)

  breaks_3 <- seq(0.5, 1.5, 0.1)
  colors_3 <- colorRampPalette(c('blue', 'white', '#FFBFBF'))(length(breaks_3))

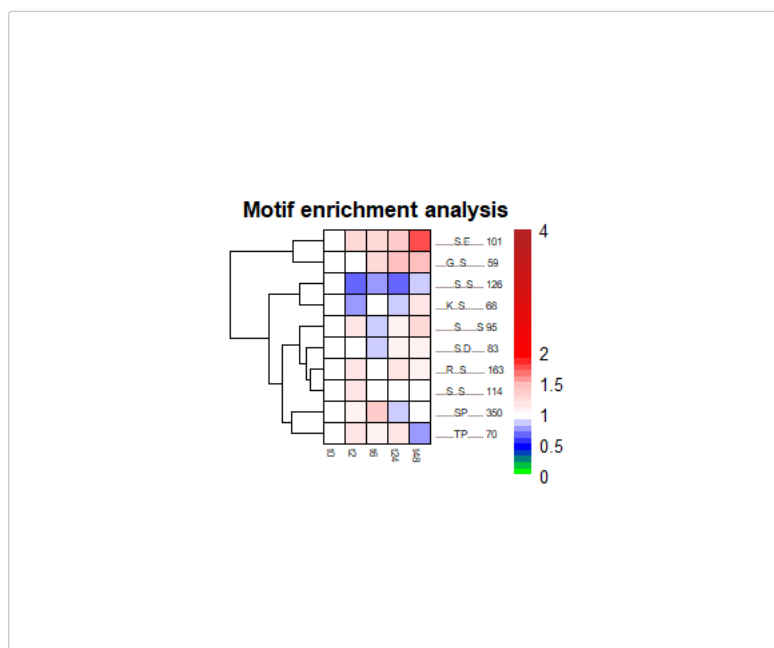
  breaks_4 <- seq(1.5, 2, 0.1)
  colors_4 <- colorRampPalette(c('#FFBFBF', 'red'))(length(breaks_4))

  breaks_5 <- seq(2, 4, 0.1)
  colors_5 <- colorRampPalette(c('red', 'firebrick'))(length(breaks_5))

  breaks <- c(breaks_1, breaks_3, breaks_4, breaks_5)
  breaks <- breaks[which(!duplicated(breaks))]
  colors <- c(colors_1, colors_3, colors_4, colors_5)
  colors <- colors[which(!duplicated(colors))]

  length(breaks)
  length(which(!duplicated(colors)))
  ph <- pheatmap(
    motif_group_m_ratio_df_mat,
    scale = 'none',
    # annotation_col = annotation_col,
    clustering_distance_cols = 'euclidean',
    fontsize_row = 6, cutree_rows = 1, show_rownames = TRUE, cluster_rows = TRUE,
    fontsize_col = 6, cutree_cols = 1, show_colnames = TRUE, cluster_cols = FALSE,
    border_color = 'black',
    # color = colors,
    cellwidth = 12, cellheight = 12,
    breaks = breaks,
    color = colors,
    legend_breaks = c(0, 0.5, 1, 1.5, 2, 4),
    legend_labels = c(0, 0.5, 1, 1.5, 2, 4),
    main = 'Motif enrichment analysis'
  )
}

```



KSEA method

- formatting output

```
# formatting output
formatted_output_df <- formatted_output_mef_results(foreground_sequences_mapped_to_motifs)

## Output formatted sequences in foreground that are mapped to specific motif.

# write file
# write.table(formatted_output_df, 'formatted_output_df.txt', row.names = FALSE, col.names = FALSE, sep
= '\t')
```

Session Info

```
sessionInfo()

## R version 3.3.2 (2016-10-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows >= 8 x64 (build 9200)
##
## locale:
## [1] LC_COLLATE=Chinese (Simplified)_China.936
## [2] LC_CTYPE=Chinese (Simplified)_China.936
## [3] LC_MONETARY=Chinese (Simplified)_China.936
## [4] LC_NUMERIC=C
## [5] LC_TIME=Chinese (Simplified)_China.936
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] ggseqlogo_0.1   pheatmap_1.0.8  PhosMap_0.99.33
##
## loaded via a namespace (and not attached):
## [1] lattice_0.20-35      samr_3.0         colorspace_1.3-2
## [4] htmltools_0.3.6      yaml_2.1.18      rlang_0.2.0
## [7] e1071_1.6-8          later_0.8.0      pillar_1.3.0
## [10] withr_2.1.2          RColorBrewer_1.1-2 matrixStats_0.53.1
## [13] foreach_1.4.4        plyr_1.8.4       GSA_1.03.1
## [16] stringr_1.3.0        munsell_0.4.3    commonmark_1.4
## [19] gtable_0.2.0         devtools_1.13.5  evaluate_0.10
## [22] codetools_0.2-15     memoise_1.1.0    labeling_0.3
## [25] knitr_1.20           BiocInstaller_1.24.0 httpuv_1.5.1
## [28] parallel_3.3.2       class_7.3-14     highr_0.6
## [31] Rcpp_0.12.16         xtable_1.8-2     backports_1.1.2
## [34] promises_1.0.1       scales_0.5.0     limma_3.30.13
## [37] desc_1.2.0           jsonlite_1.5     mime_0.5
## [40] ClueR_1.4            shinyFiles_0.7.3  fs_1.3.1
## [43] impute_1.48.0        ggplot2_2.2.1    digest_0.6.14
## [46] stringi_1.1.7        openxlsx_4.0.17  Rtsne_0.13
## [49] shiny_1.3.2          rprojroot_1.3-2  grid_3.3.2
## [52] tools_3.3.2          bitops_1.0-6     magrittr_1.5
## [55] lazyeval_0.2.1       RCurl_1.95-4.10  glmnet_2.0-16
## [58] tibble_1.4.2         crayon_1.3.4     Matrix_1.2-14
## [61] xml2_1.2.0           rmarkdown_1.4    assertthat_0.2.0
## [64] roxygen2_6.0.1       rstudioapi_0.7   iterators_1.0.8
## [67] R6_2.2.2
```