My Thesis

STK analysis - Without Chip 19, Male Vehicle and Male Insulin

Marziyeh S Jahromi

8/15/23

Table of contents

1 Introduction

1.1 Background

The Pamstation12 instrument provides a profiling of kinase activity of cell or tissue samples. The device is loaded with either serine/threonine or tyrosine microarray chips. Each chip has 4 wells so four samples can be loaded on a single chip, and the Pamstation12 can accommodate 3 chips per run. The microarray represents 144 (STK chip) or 196 (PTK chip) reporter peptides that can be phosphorylated by serine/threonine or tyrosine kinases. The device measures the degree of the phosphorylation in real time by detecting fluorescently labeled antibodies at different exposure times. The list of peptides present in each microarray can be viewed here: STK chip, PTK chip

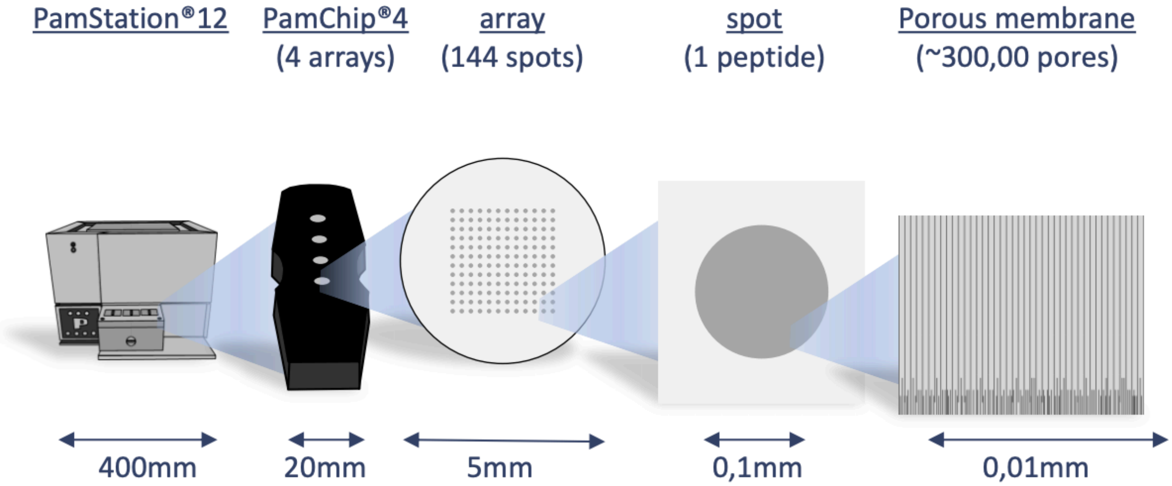


Figure 1.1: Pamgene Kinase Activity Platform

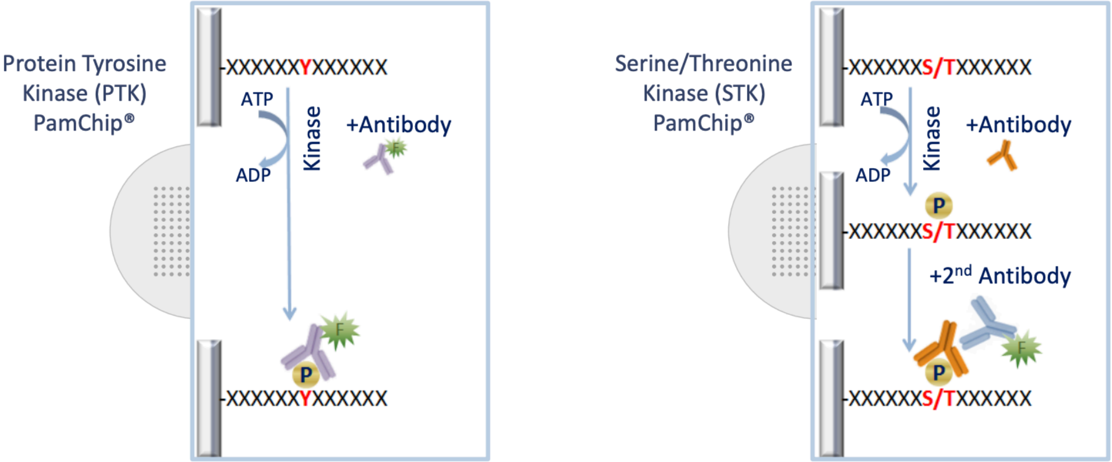
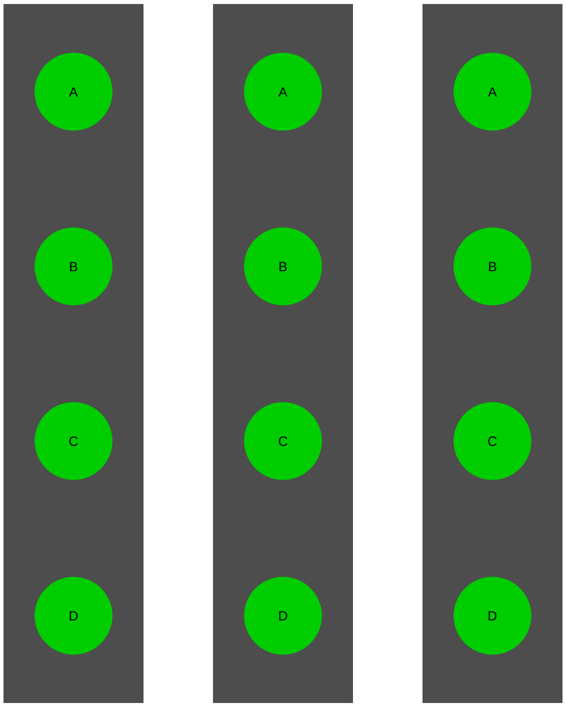


Figure 1.2: Pamgene Platform Detection

2 Run Design

Designing the placement of the samples on the chips and arrays is important to consider due to the variability across different chips and batches. During the run some wells are subject to fail and their data cannot be analyzed and shown below as red.



3 Results

3.1 Image Analysis

The first step of analyzing the run is to convert the images taken by the PamStation of each array at different exposure times to numerical values This is done by the Bionavigator software developed by Pamgene. The software recognizes the grid of the array with the aid of the searching algorithm (Pamgrid) to correctly identify each spot on the array. The numbers produced by this software represent the median value of the foreground pixels minus the median value of the background pixels to produce the median signal minus background (Median_SigmBg).

3.2 Reading Data

The first step will be reading the crosstab view bionavigator files (Median_SigmBg and Signal_Saturation) and defining the PamChip type (STK or PTK). The raw data is read and then transformed to be in tidy format for an easier analysis, modeling, and visualizing.

3.3 QC Initial Steps and Groups Assignments

We will perform a couple of quality control steps to deal with negative values in the data and adjust based on signal saturation (optional). Next, we will define a new column to represent the grouping. And then, we will extract end point signal values

3.4 QC Steps and Model Fitting

We will filter out peptides with low signals. In order to combine the values from different exposure times into a single value, a simple linear regression model of the *Median_SigmBg* as a function of exposure time is fitted. The slope of of the model fit and R^2 are then used for quality control and samples comparison. The slope is also scaled by multiplying by 100 and log2 transformed (*Slope_Transformed*). We then filter out peptides with poor linear fit and references peptides.

3.5 Global Signal Intensity

For a global signal intensity across all samples/groups, few figures can be plotted based on the *Slope_Transformed* values.

3.5.1 Global CV Plots

We will plot the coefficient of variation on both the normal and normalized fits. This will help us to identify groups with high variation that could be explained by sample outliers.

3.5.2 Global Violin Plots

We will plot violin figures to examine global signal differences between groups/samples.

3.5.3 Global Heatmaps

The heatmap represent all the peptides present on the chip except the positive/internal controls and peptides that failed to pass QC. The heatmaps are scaled by row to highlight the peptide signal differences across the samples. A hierarchical unsupervised clustering is applied both on the peptides and the samples to group potentially similar signatures.

3.6 Group Comparison

To compare between samples, a two-group comparison is performed. In this case, there are three group comparisons:

- AMPK KD Neuron + Untreated vs Wild Type Neuron + AICAR
- AMPK KD Neuron + Untreated vs Wild Type Neuron + Untreated
- AMPK KD Neuron + AICAR vs Wild Type Neuron + AICAR

The *Slope_Transformed* ratio between each group, paired by chip, is calculated to the fold change. Based on the fold change, peptides that pass a certain fold change threshold are considered significant hits. Also, quality control steps applied in each comparison to filter out peptides that do not reach specific criteria:

- The *Median_SigmBg* at max exposure *100ms* must be above a certain value

- R^2 of the linear model fit must be above a threshold value

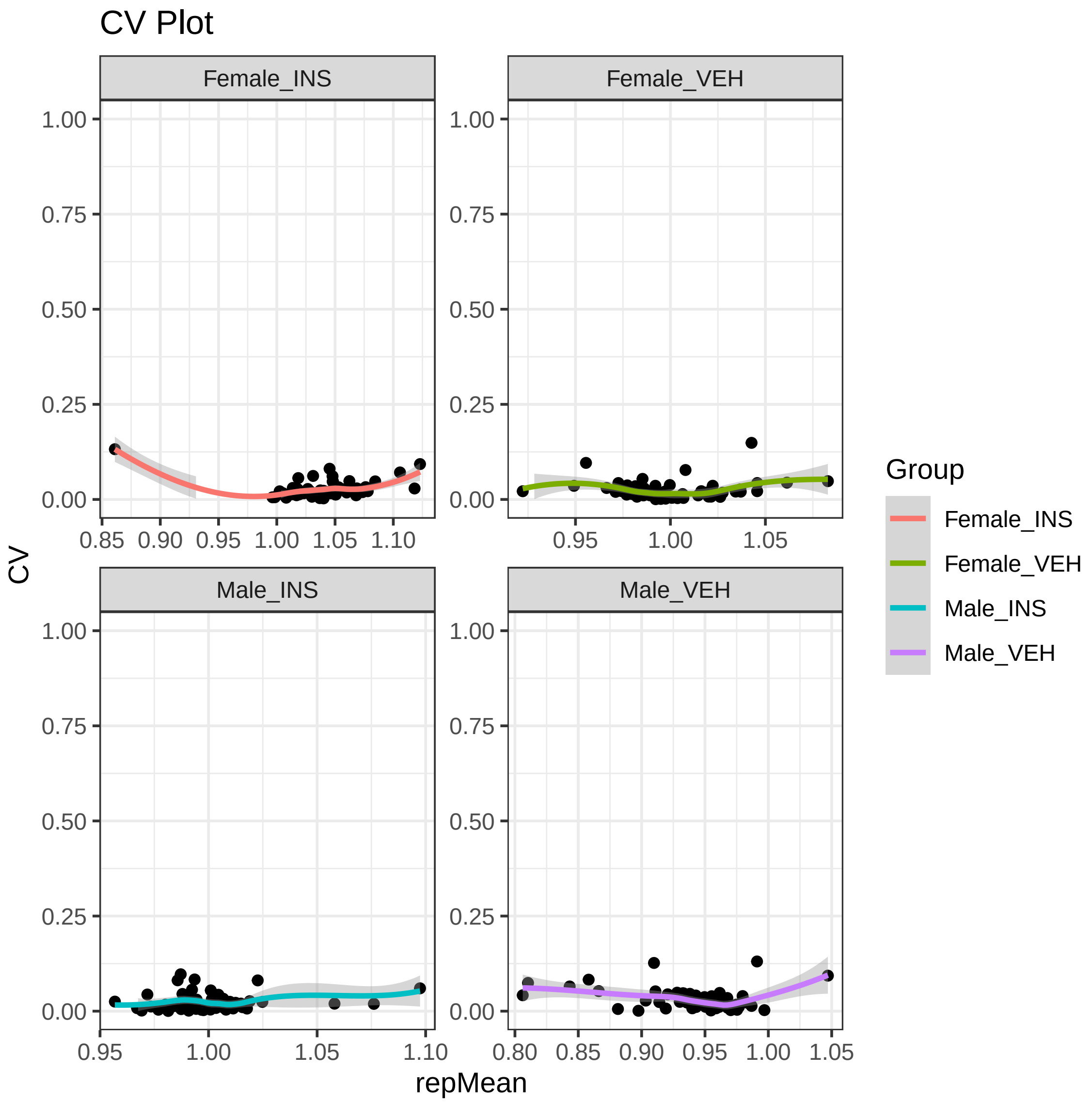


Figure 3.1: Coefficient of Variation plotted for each peptide across all 4 groups

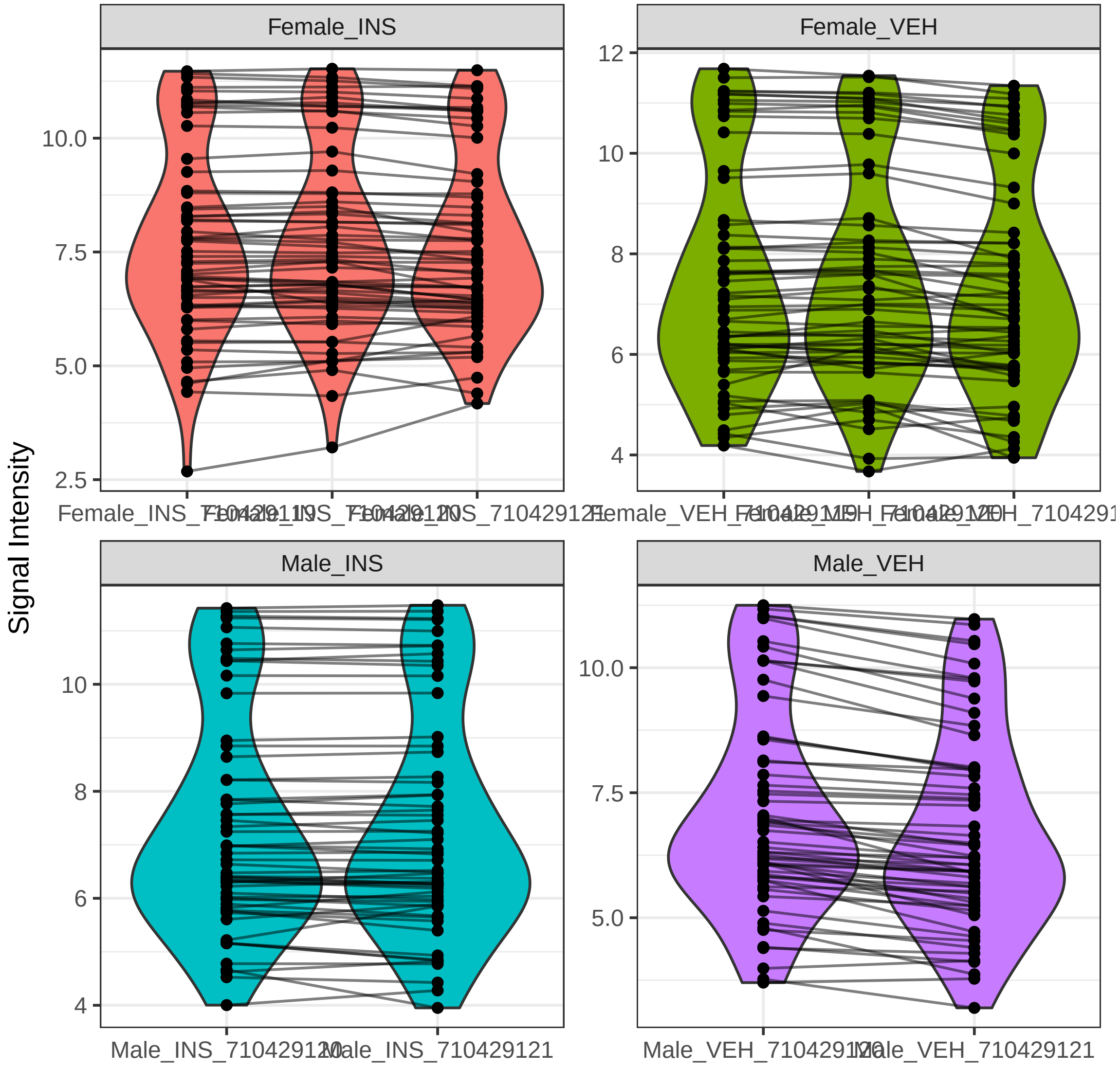


Figure 3.2: Violin Plots for signal intensity Distribution Across Groups for all replicates

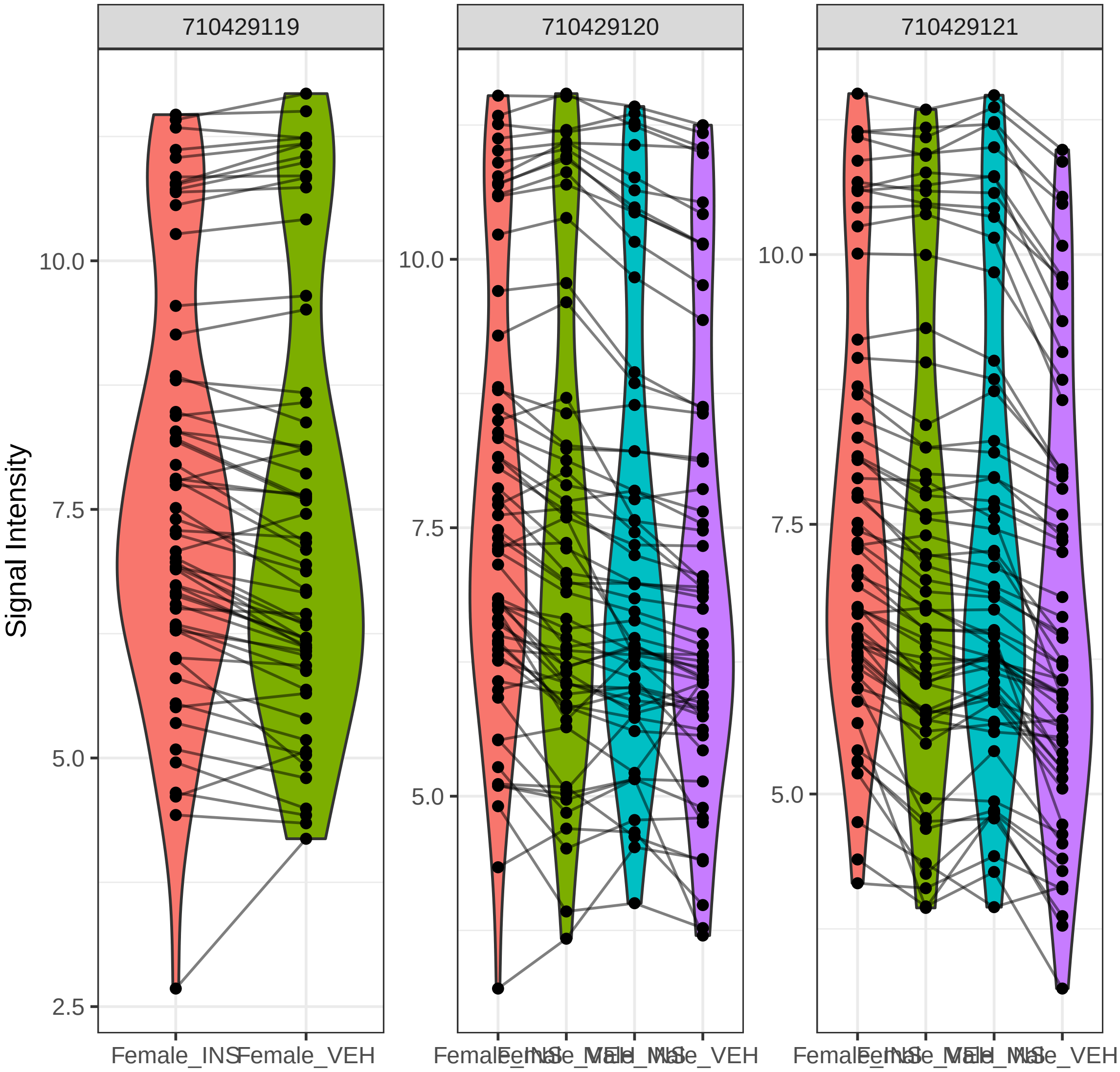


Figure 3.3: Violin Plots for signal intensity Distribution Across Chips for all replicates

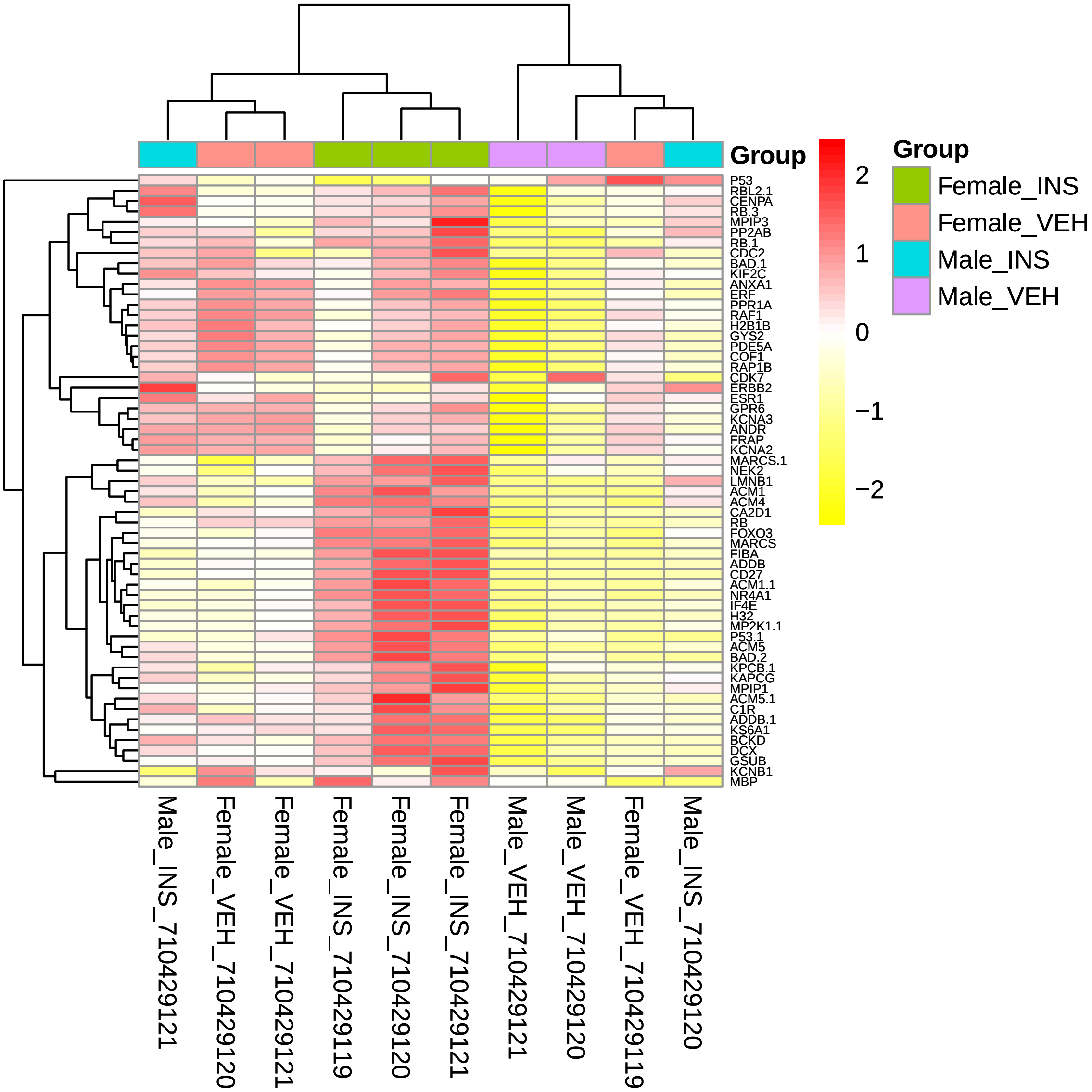


Figure 3.4: Row and chip normalized intensity values for the selected peptides

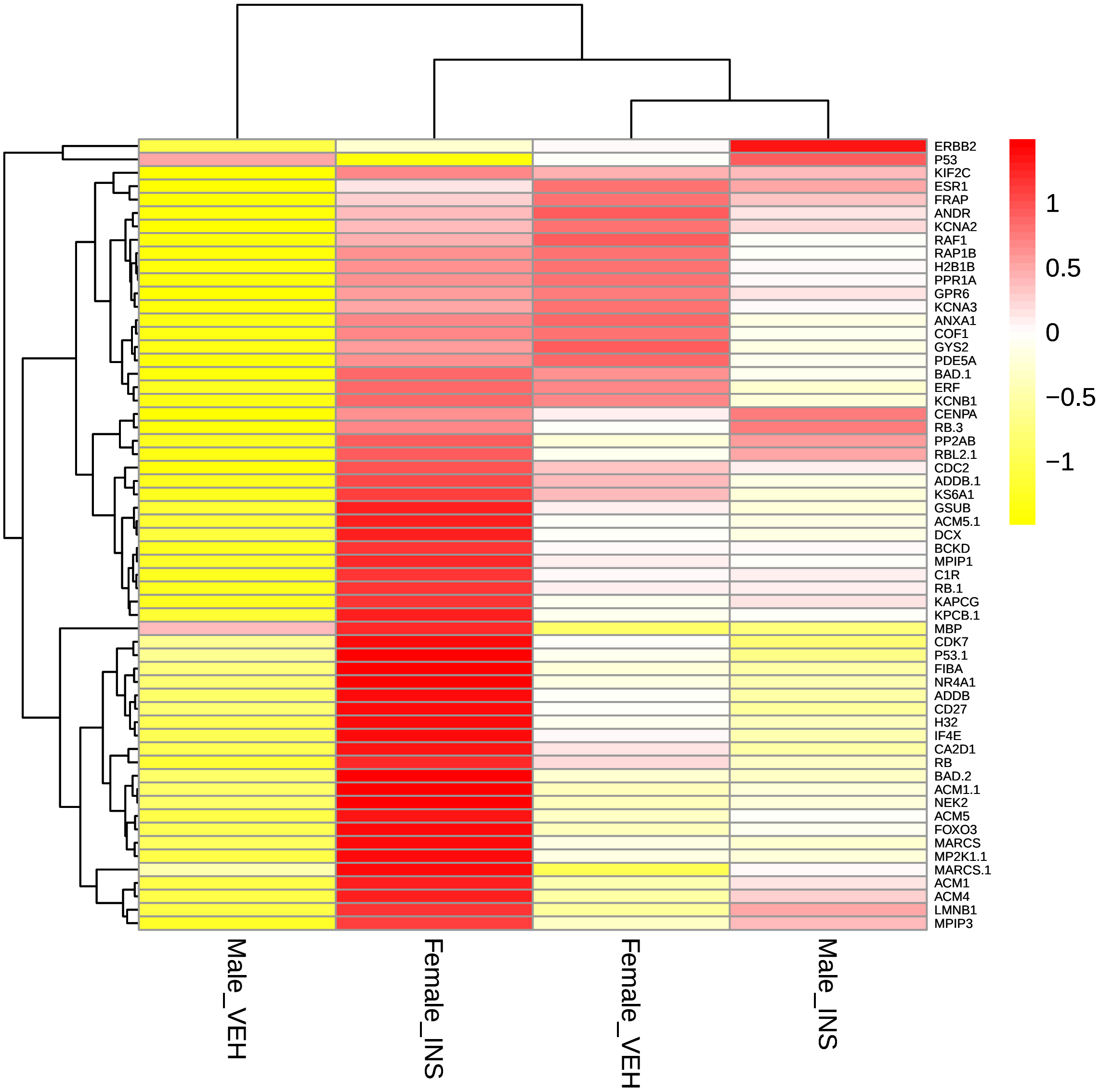
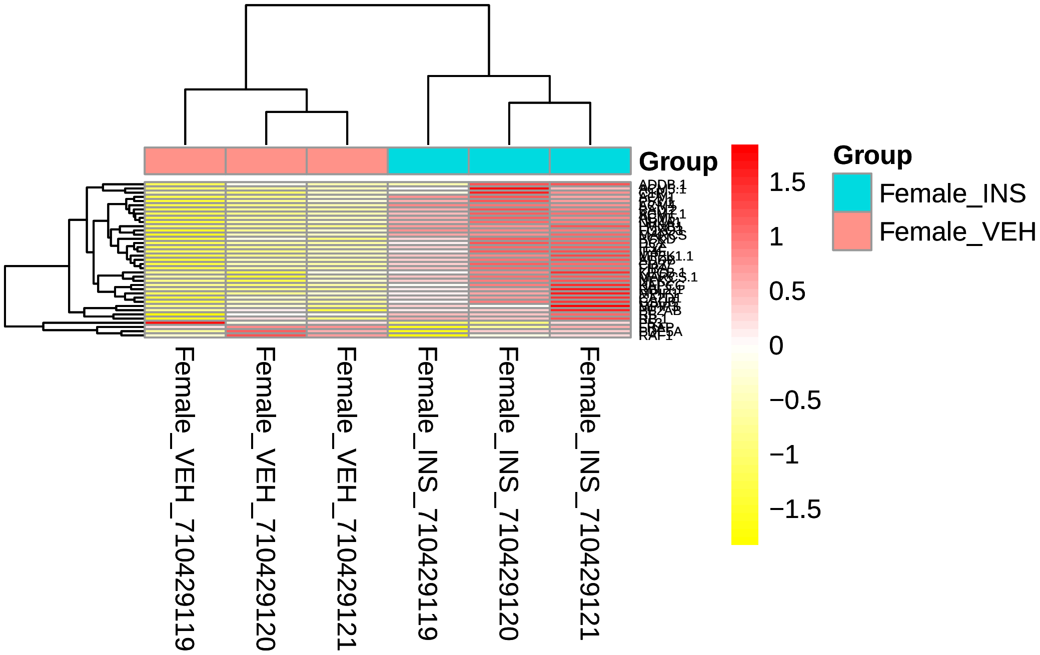


Figure 3.5: Row and group normalized intensity values for the selected peptides

These *Filtering Parameters* (fold change threshold, QC criteria) can be modified to adjust the stringency of the analysis. The *Filtering Parameters* that are used for this analysis:

- The *Median_SigmBg* at max exposure *100ms* must be equal or above 5
- R^2 of the linear model fit must be above or equal 0.8
- Log fold change (LFC) cutoffs at (0.2,0.3,0.4)



3.6.1 Female_INS vs Female_VEH

3.6.1.1 Heatmap

After applying the *Filtering Parameters* for this group comparison, only *37*/141 peptides carried forward in the analysis (i.e. *37 hits*). Below are some figures to visualize the differences between these samples for considering these *hits*.

3.6.1.2 Violin Plot

Below, the violin plot visualizes the distribution of selected peptides for the analysis.

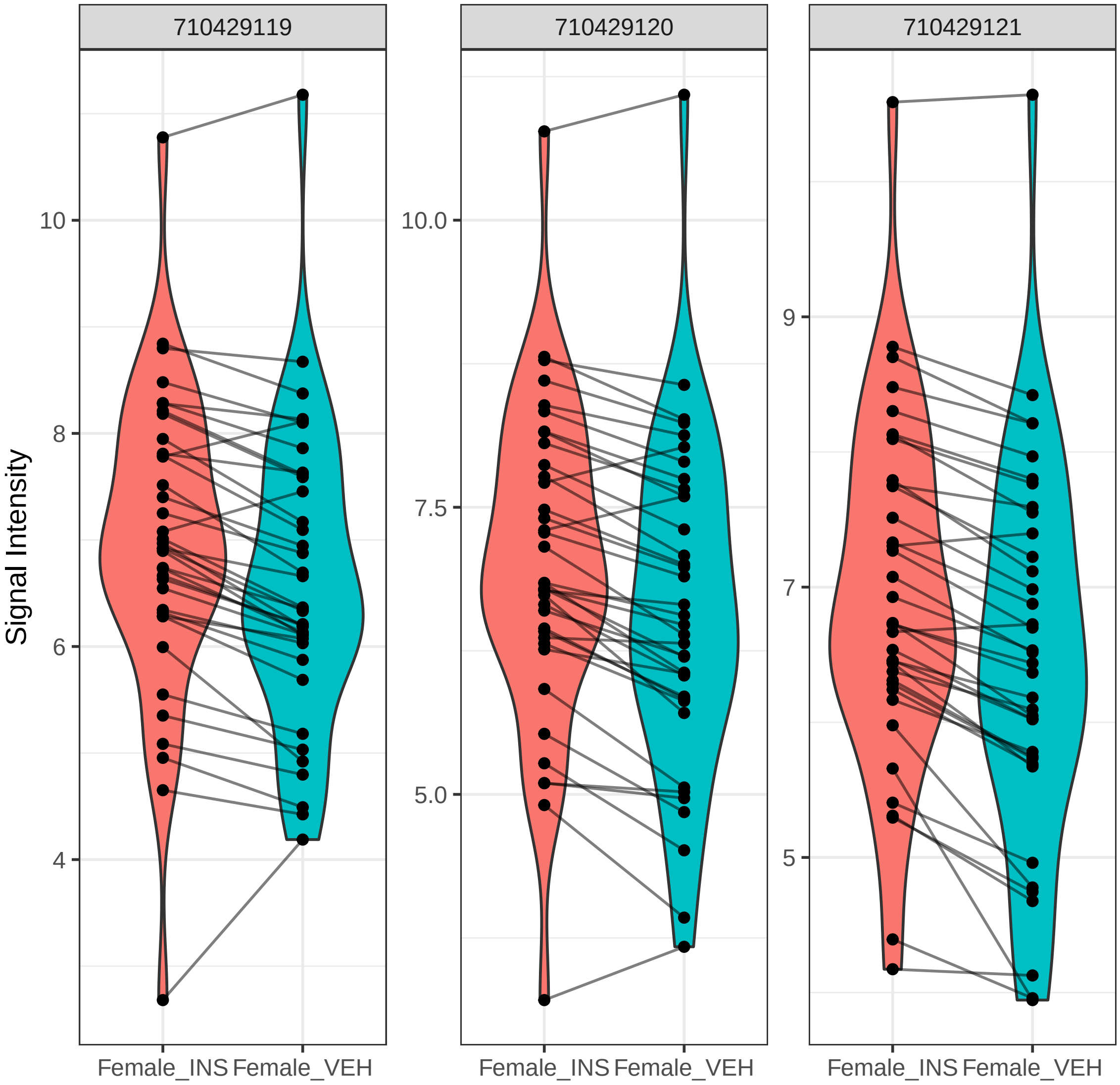


Figure 3.6: Violin plot of two groups

3.6.1.3 Waterfall Plot

This waterfall represents the log2 fold changes between the two groups at each peptide.

3.6.1.4 Upstream Kinase Analysis

The lab carefully curated and mapped the kinases that can act and phosphorylate each peptide present on the chip. This was achieved by using multiple sources including GPS 3.0, Kinexus Phosphonet, PhosphoELM and PhosphoSite Plus. Based on that association between peptides and kinases, a random sampling analysis is performed for these hits. The basic idea of *KRSA* is: For each iteration (*2000* iterations performed in this analysis), the same number of hits are randomly selected from the total 141/or 193 peptides present on the chip. Predicted kinases are then mapped to this sample list of peptides and number of kinases are determined. The kinase count from the actual hits and random sampling is then compared to determine the significance.

Kinase	AvgZ
PDHK	-5.865833
PDK1	-6.045000
PIM	-6.665000
QIK	-6.710833
RSK	-8.005833
PHK	-9.855833
PEK	-9.903333
CDK	-10.381667
DMPK	-10.980000
PKCA	-16.359167

Method	NumberOfPeptides
meanLFC.0.15	43
meanLFC.0.2	37
meanLFC.0.3	31
meanLFC.0.4	23
710429119.0.15	42
710429119.0.2	41
710429119.0.3	32
710429119.0.4	24
710429120.0.15	42
710429120.0.2	40
710429120.0.3	30

Method	NumberOfPeptides
710429120.0.4	22
710429121.0.15	42
710429121.0.2	39
710429121.0.3	31
710429121.0.4	23

3.6.1.5 Z Scores Plot

We will plot the individual and averaged Z scores using both the across and within chip analyses.

3.6.1.6 Reverse KRSA Plot

We will use the reverse KRSA plot function, to plot the log2 fold change values for all peptides mapped to kinase hits. This will help us examine the activity of the kinase

3.6.1.7 Coverage Plot

To view the coverage of kinases across the full list of peptides on the chip, we will use the coverage plot function

3.6.1.8 Ball Model Network

We will view the ball model network function, to generate a model representing the protein-protein interactions between kinases

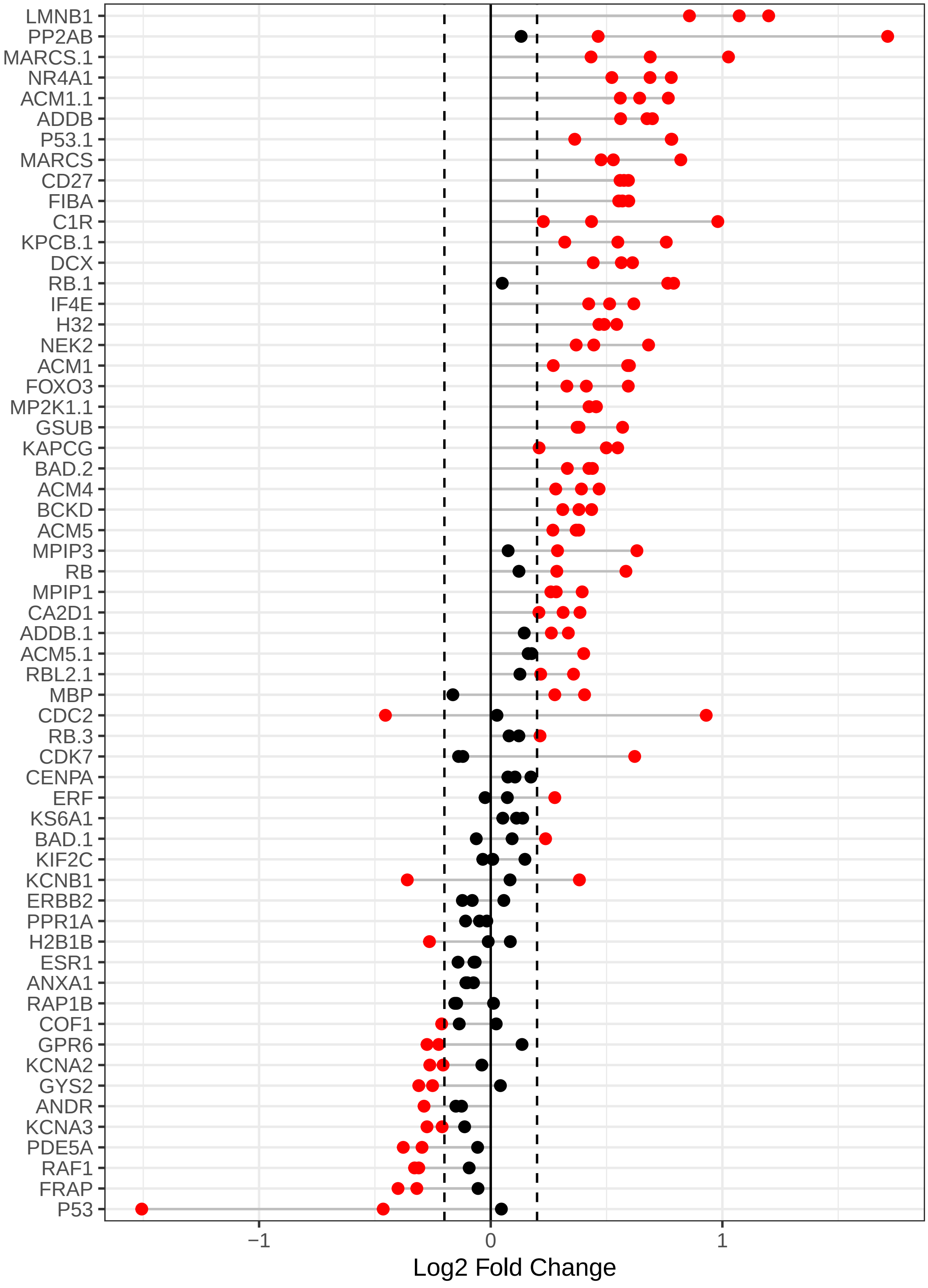


Figure 3.7: Waterfall Plot to show the distribution of change in peptide phosphorylation

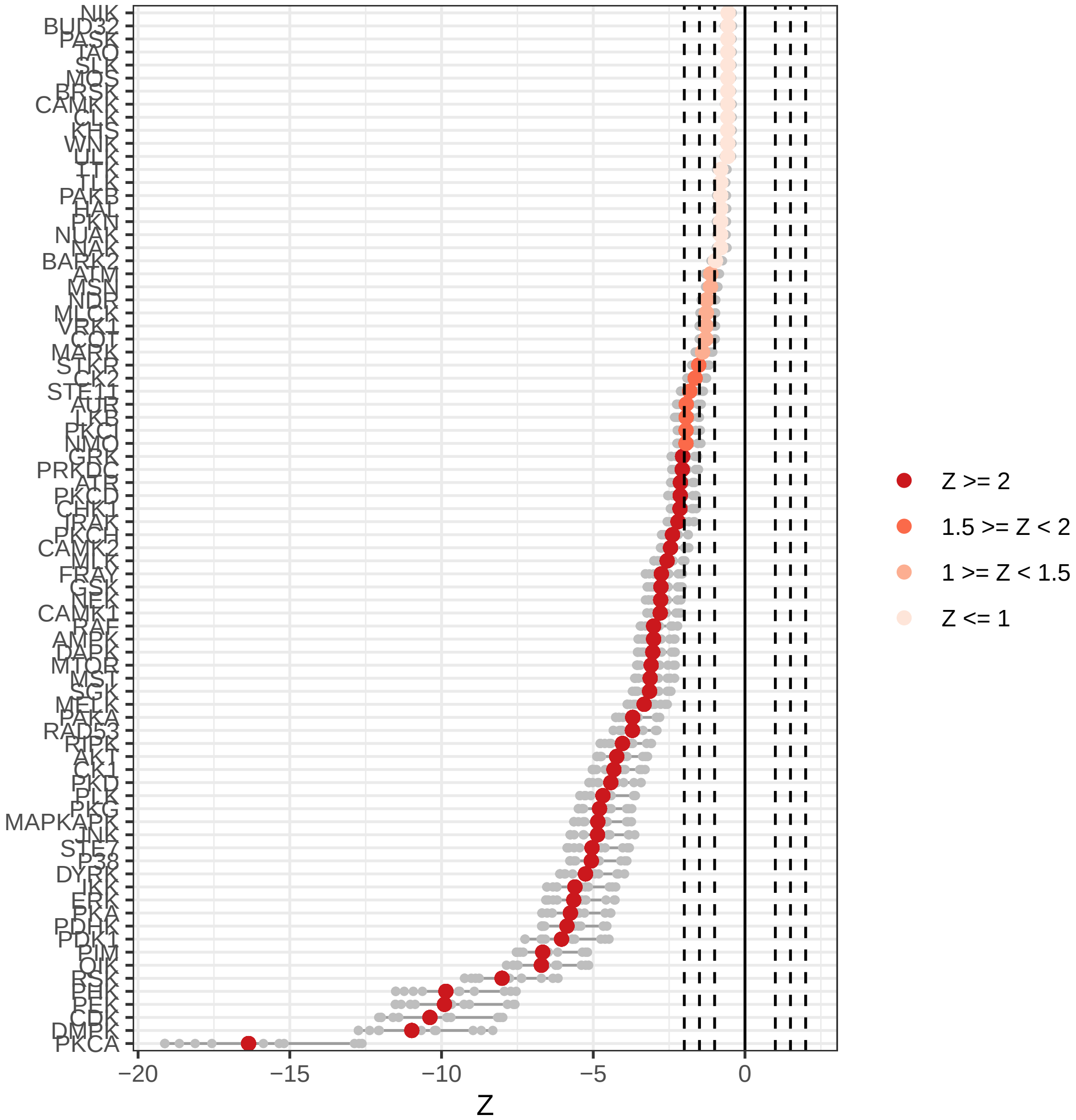


Figure 3.8: Waterfall plot of the Z Scores of each kinase family

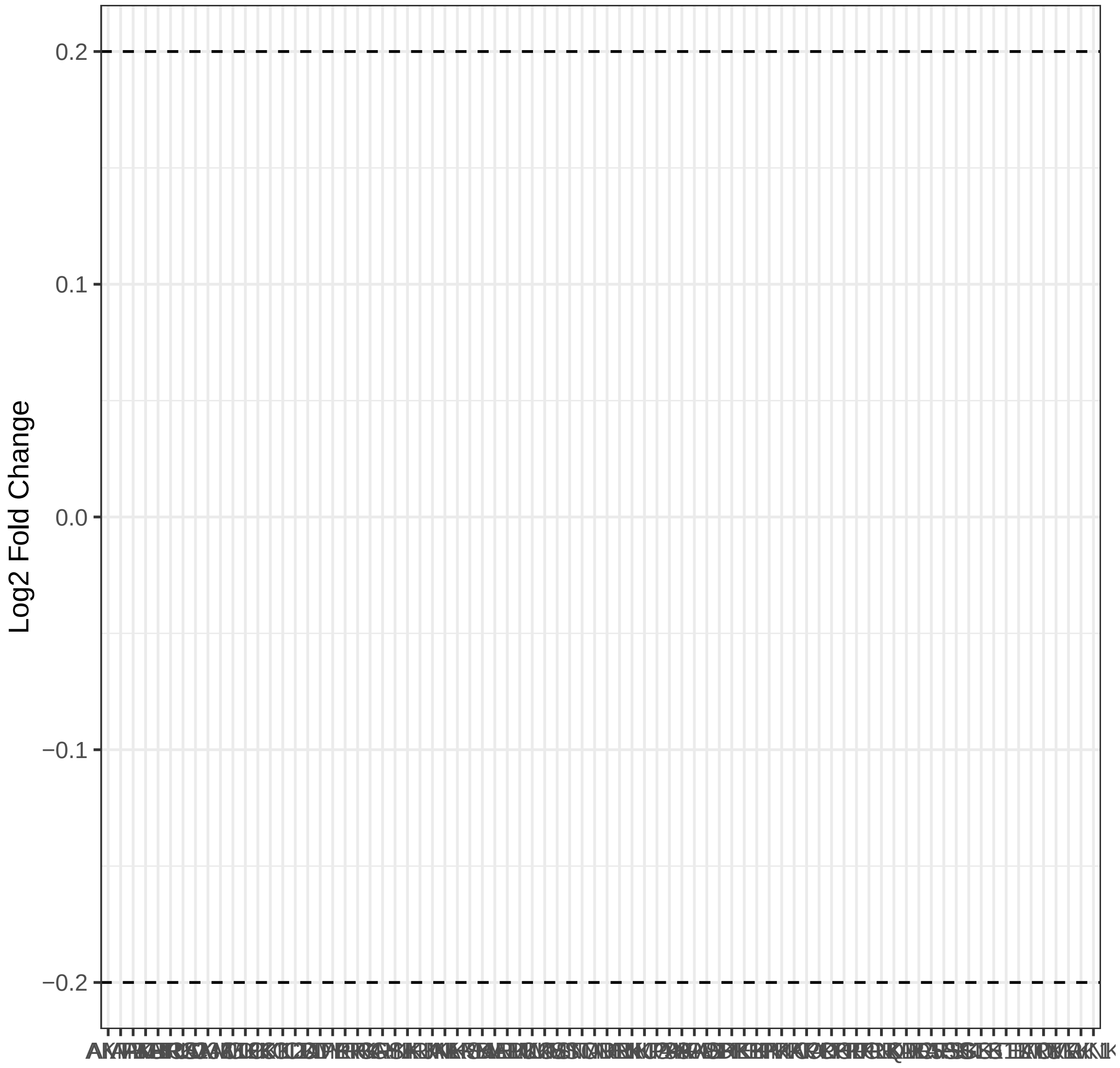


Figure 3.9: Kinase Activity summary for each kinase family based on peptide phsophorylation

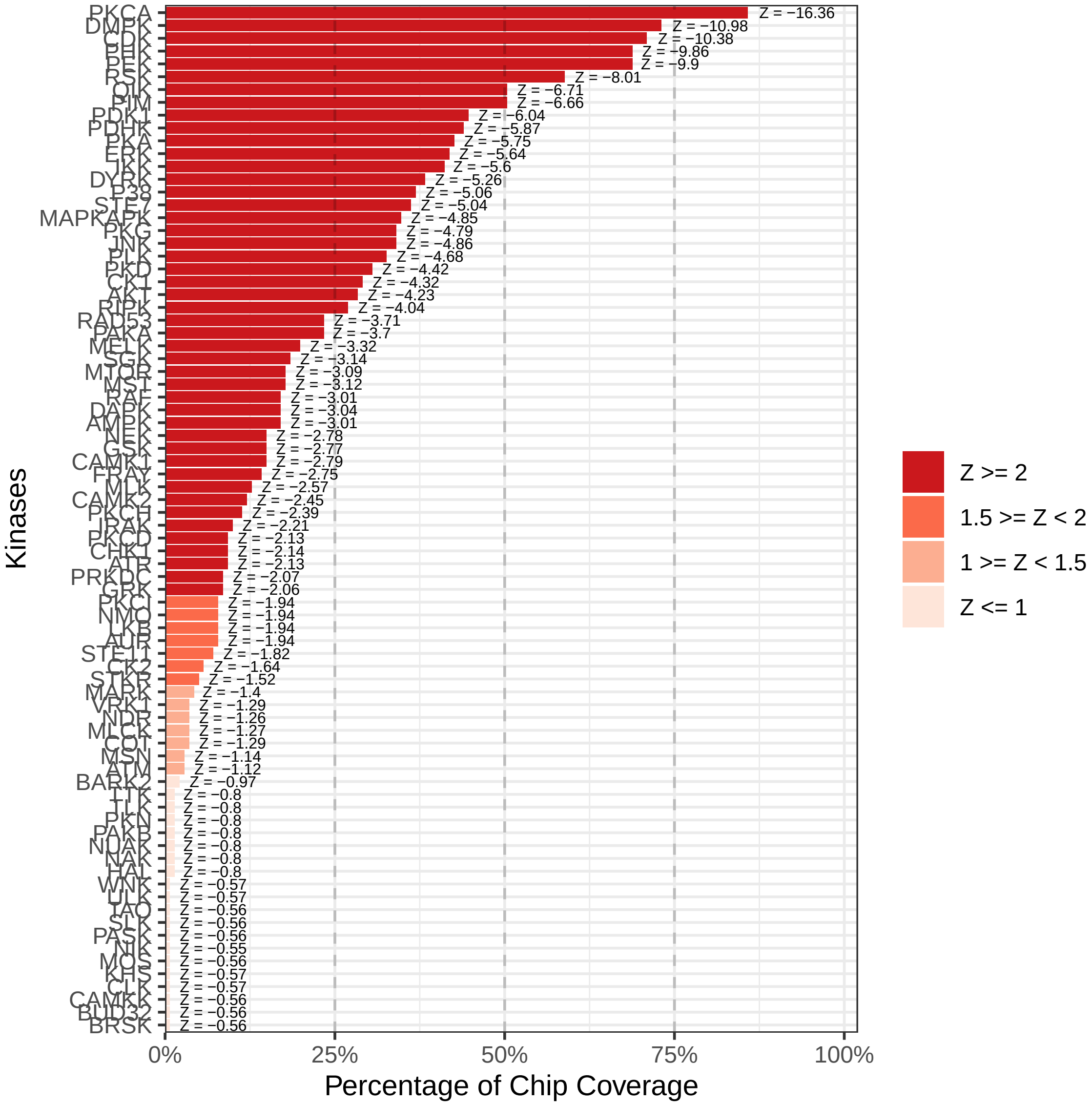
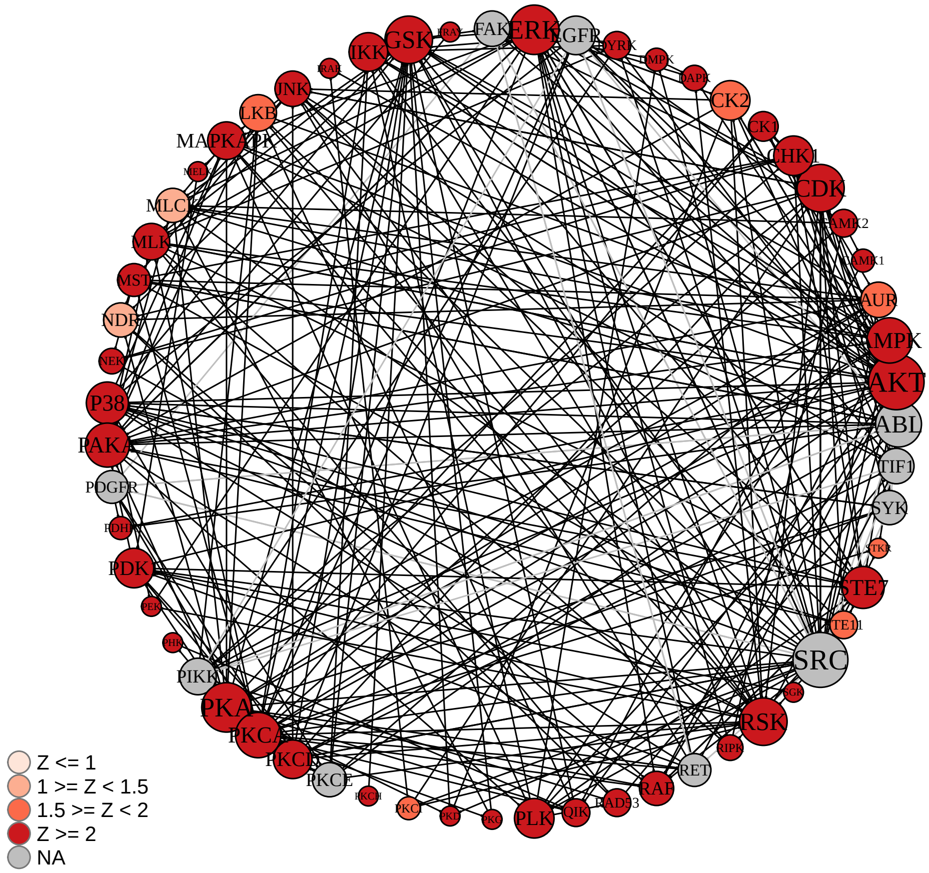
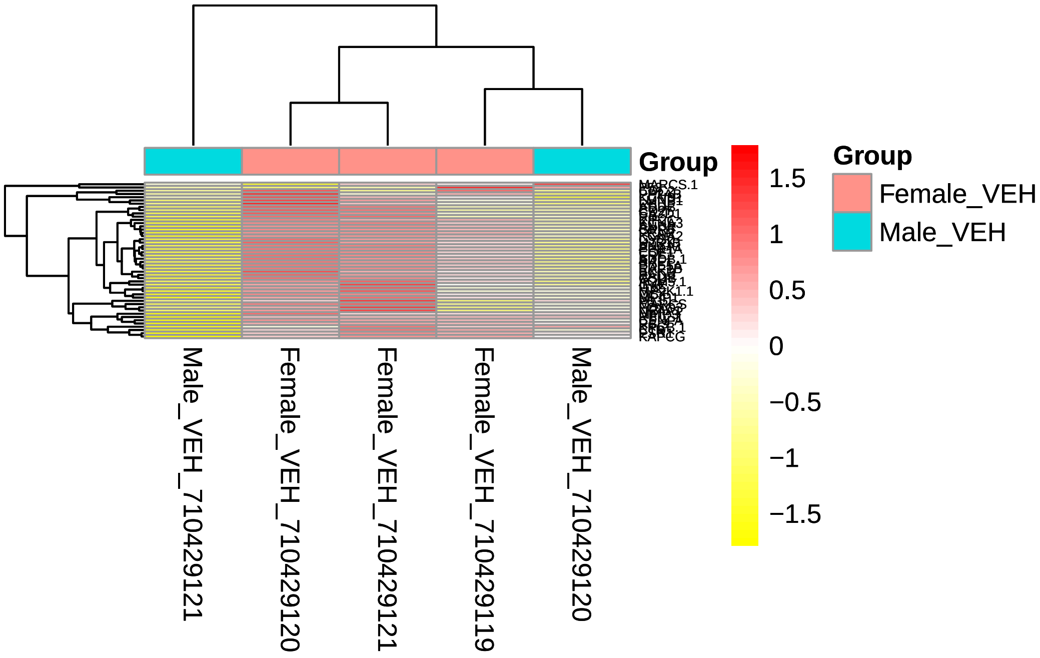


Figure 3.10: Percentge of peptides each kinase family phosphorylates


ERK
EGFR
DYRK
DMPK
DAPK
CK2
CK1
CHK1
CDK
CAMK2
CAMK1
AUR
AMPK
AKT
ABL
TIF1
SYK
STKR
STE7
STE11
SRC
SGK
RSK
RIPK
RET
RAF
RAD53
QIK
PLK
PKG
PKD
PKCI
PKCH
PKCE
PKCD
PKCA
PKA
PIKK
PHK
PEK
PDK1
PDHK
PDGFR
PAKA
P38
NEK
NDR
MST
MLK
MLCK
MELK
MAPKAPK
LKB
JNK
IRAK
IKK
GSK
FRAY
FAK
Z <= 1
1 >= Z < 1.5
1.5 >= Z < 2
Z >= 2
NA



3.6.2 Male_VEH vs Female_VEH

3.6.2.1 Heatmap

After applying the *Filtering Parameters* for this group comparison, only *48*/141 peptides carried forward in the analysis (i.e. *48 hits*). Below are some figures to visualize the differences between these samples for considering these *hits*.

3.6.2.2 Violin Plot

Below, the violin plot visualizes the distribution of selected peptides for the analysis.

3.6.2.3 Waterfall Plot

This waterfall represents the log2 fold changes between the two groups at each peptide.

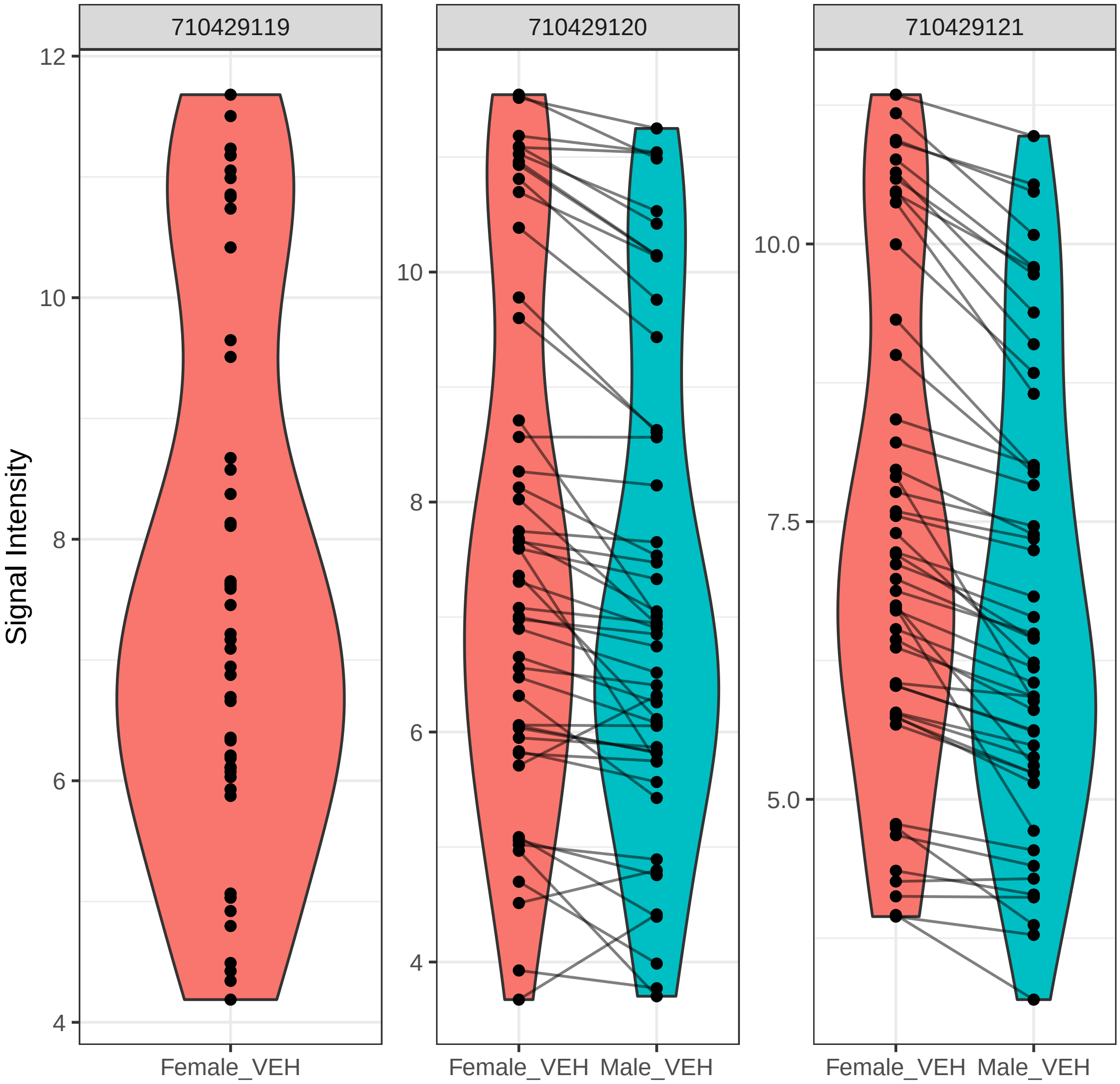


Figure 3.11: Violin plot of two groups

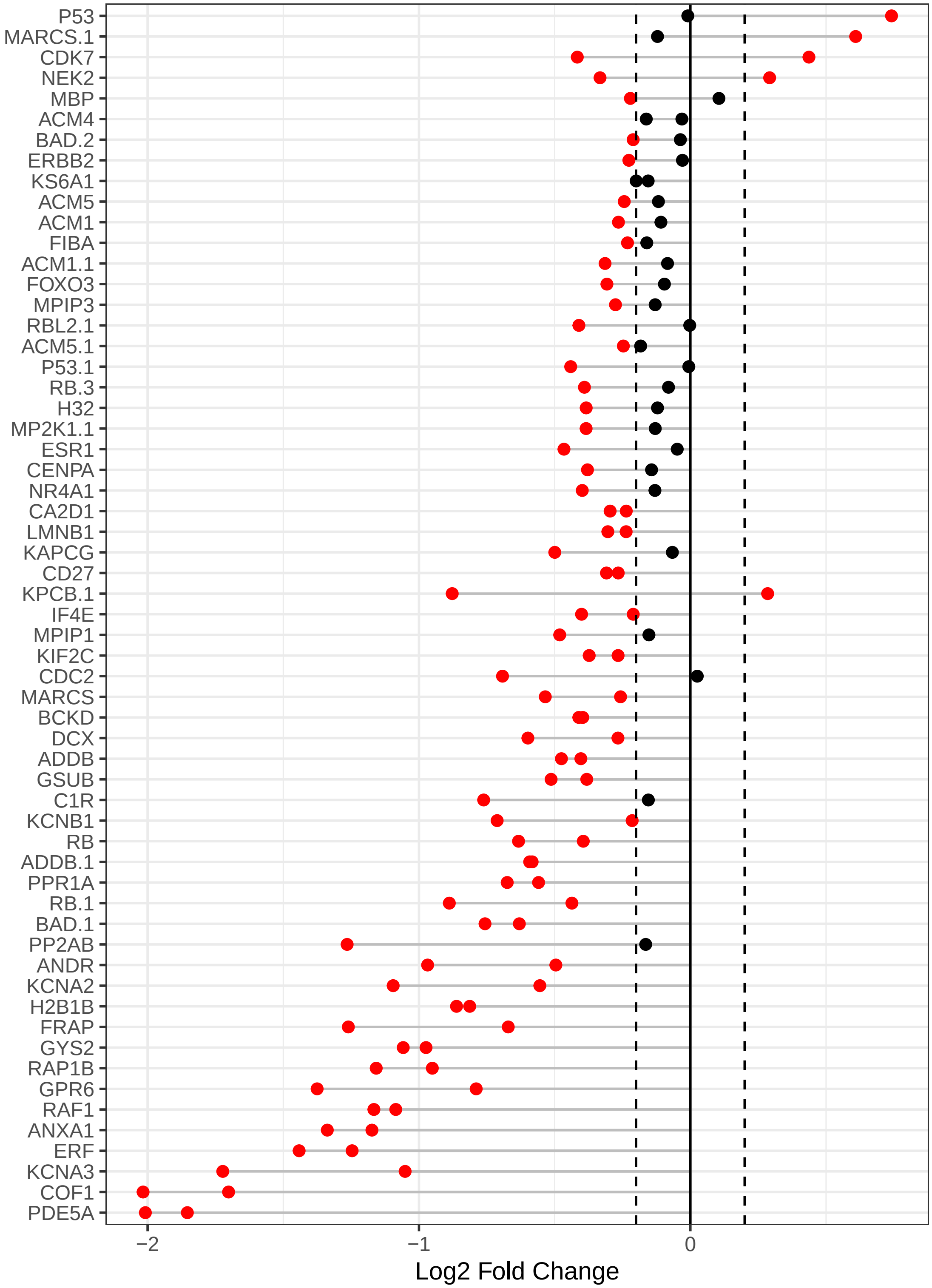


Figure 3.12: Waterfall Plot to show the distribution of change in peptide phosphorylation

3.6.2.4 Upstream Kinase Analysis

The lab carefully curated and mapped the kinases that can act and phosphorylate each peptide present on the chip. This was achieved by using multiple sources including GPS 3.0, Kinexus Phosphonet, PhosphoELM and PhosphoSite Plus. Based on that association between peptides and kinases, a random sampling analysis is performed for these hits. The basic idea of *KRSA* is: For each iteration (*2000* iterations performed in this analysis), the same number of hits are randomly selected from the total 141/or 193 peptides present on the chip. Predicted kinases are then mapped to this sample list of peptides and number of kinases are determined. The kinase count from the actual hits and random sampling is then compared to determine the significance.

Kinase	AvgZ
PDHK	-6.44250
PDK1	-6.69375
PIM	-7.31375
QIK	-7.35375
RSK	-8.76375
PHK	-10.90000
PEK	-10.95000
CDK	-11.37625
DMPK	-12.09000
PKCA	-18.05750

Method	NumberOfPeptides
meanLFC.0.15	53
meanLFC.0.2	48
meanLFC.0.3	31
meanLFC.0.4	25
710429120.0.15	42
710429120.0.2	38
710429120.0.3	28
710429120.0.4	24
710429121.0.15	55
710429121.0.2	51
710429121.0.3	42
710429121.0.4	32

3.6.2.5 Z Scores Plot

We will plot the individual and averaged Z scores using both the across and within chip analyses.

3.6.2.6 Reverse KRSA Plot

We will use the reverse KRSA plot function, to plot the log2 fold change values for all peptides mapped to kinase hits. This will help us examine the activity of the kinase

3.6.2.7 Coverage Plot

To view the coverage of kinases across the full list of peptides on the chip, we will use the coverage plot function

3.6.2.8 Ball Model Network

We will view the ball model network function, to generate a model representing the protein-protein interactions between kinases

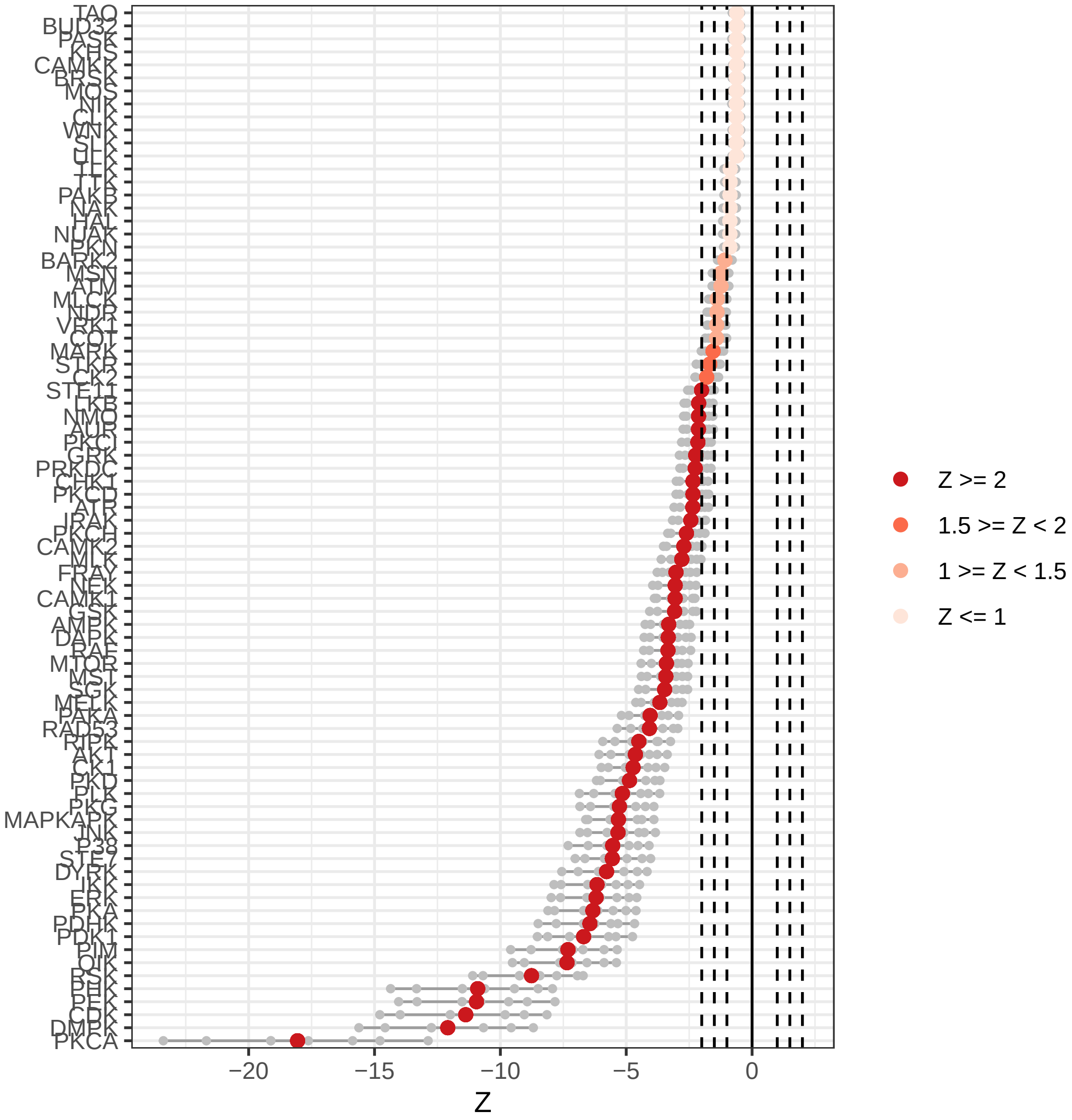


Figure 3.13: Waterfall plot of the Z Scores of each kinase family

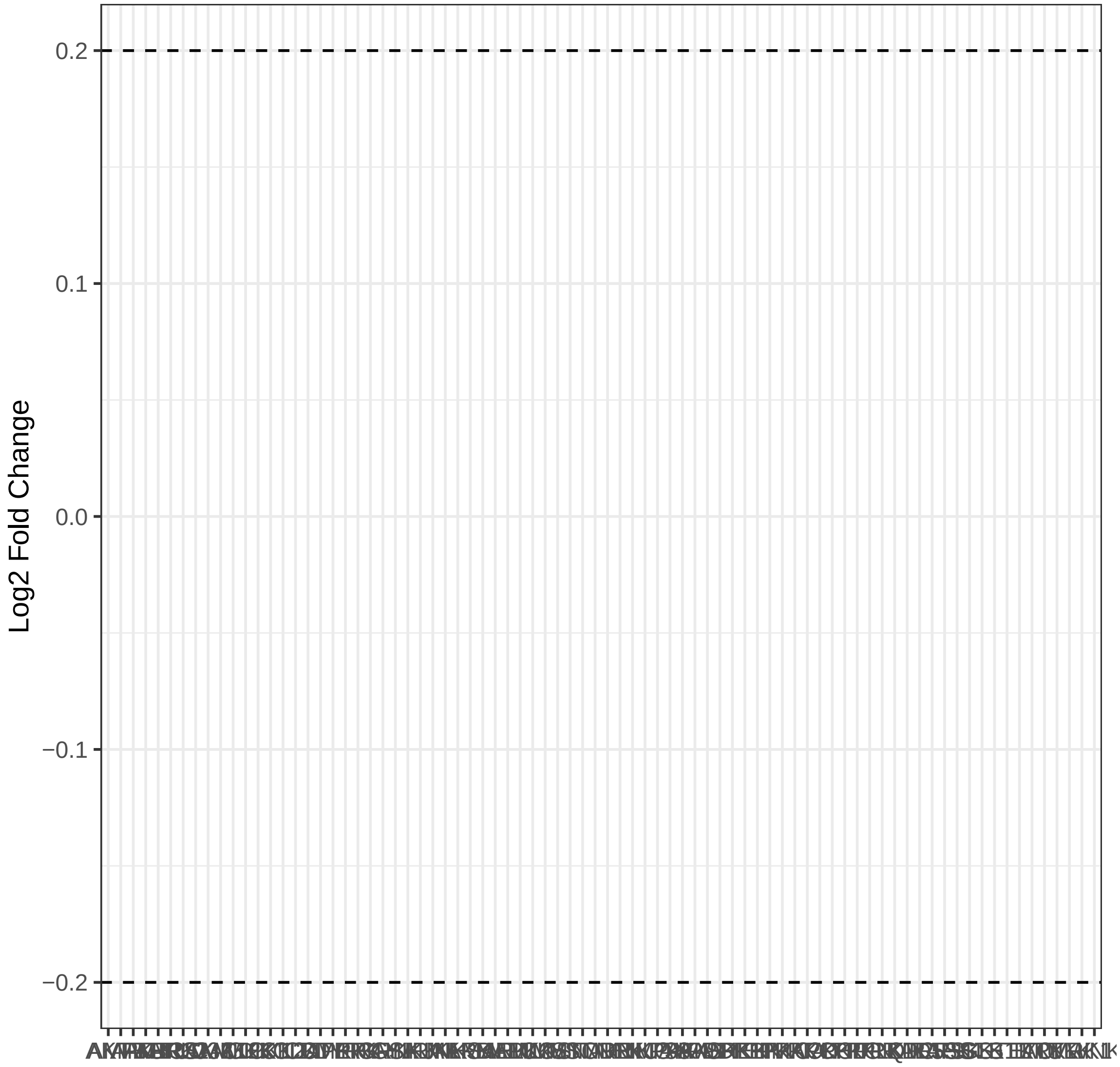


Figure 3.14: Kinase Activity summary for each kinase family based on peptide phsophorylation

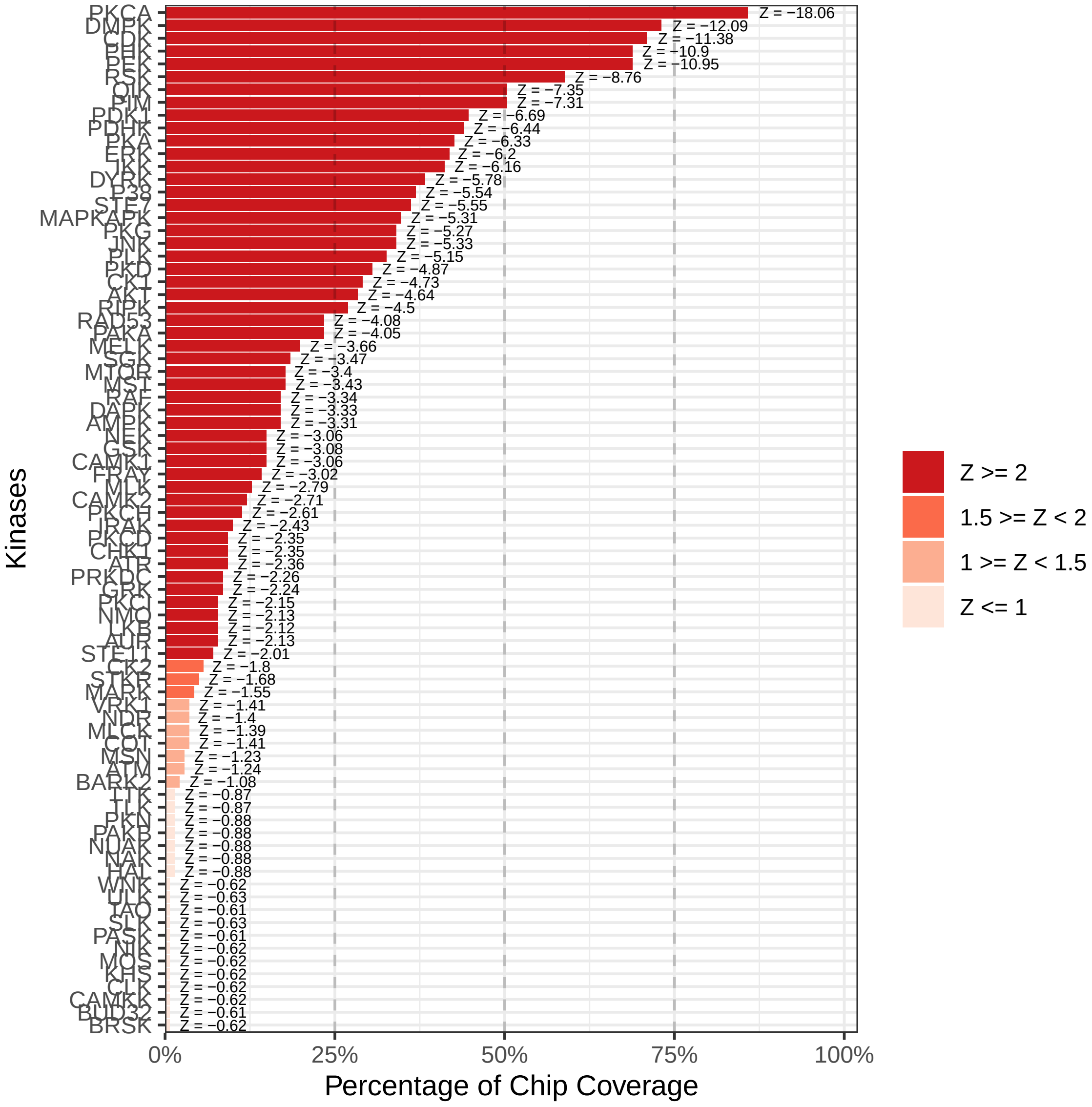
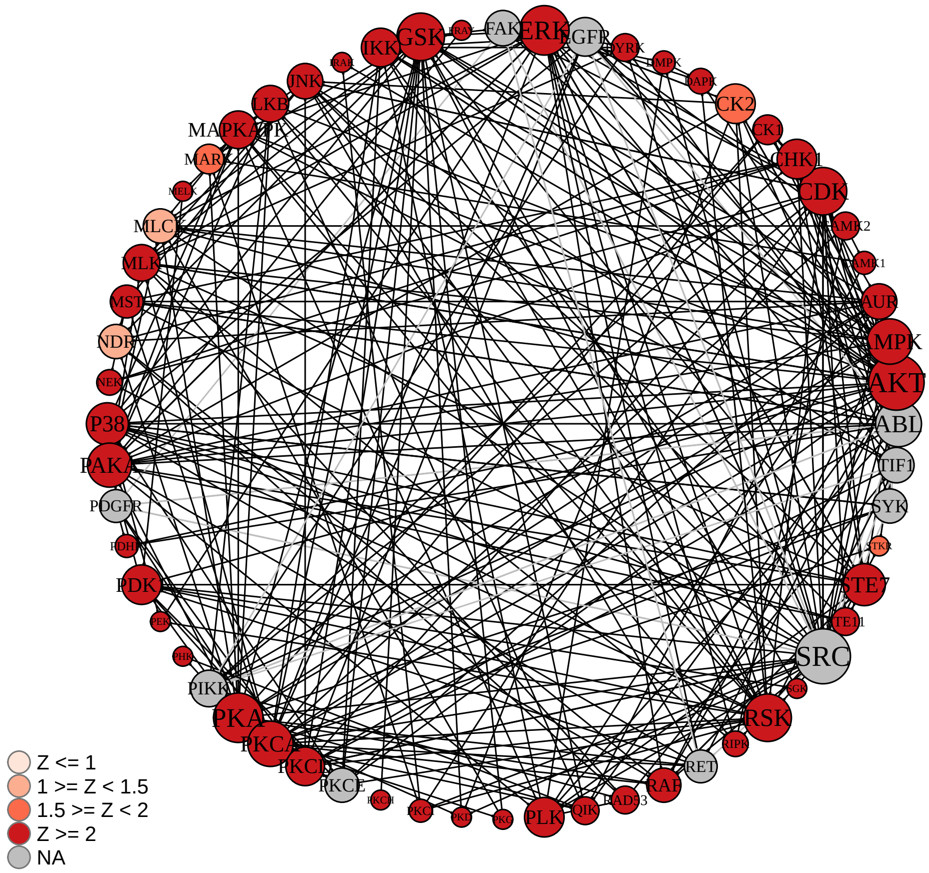
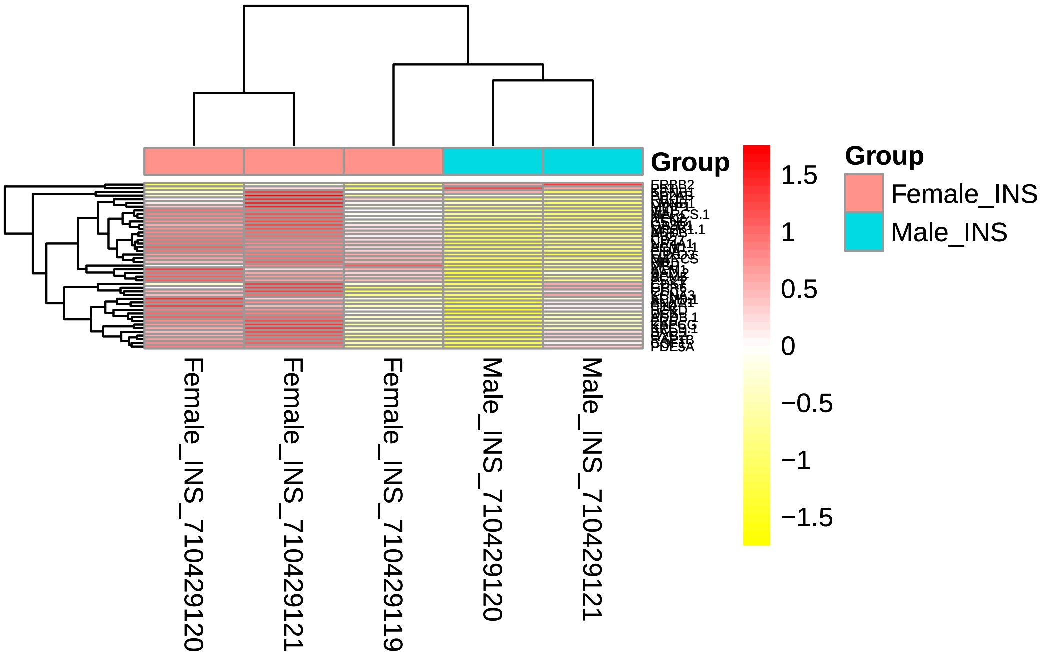


Figure 3.15: Percentge of peptides each kinase family phosphorylates


ERK
EGFR
DYRK
DMPK
DAPK
CK2
CK1
CHK1
CDK
CAMK2
CAMK1
AUR
AMPK
AKT
ABL
TIF1
SYK
STKR
STE7
STE11
SRC
SGK
RSK
RIPK
RET
RAF
RAD53
QIK
PLK
PKG
PKD
PKCI
PKCH
PKCE
PKCD
PKCA
PKA
PIKK
PHK
PEK
PDK1
PDHK
PDGFR
PAKA
P38
NEK
NDR
MST
MLK
MLCK
MELK
MARK
MAPKAPK
LKB
JNK
IRAK
IKK
GSK
FRAY
FAK
Z <= 1
1 >= Z < 1.5
1.5 >= Z < 2
Z >= 2
NA



3.6.3 Male_INS vs Female_INS

3.6.3.1 Heatmap

After applying the *Filtering Parameters* for this group comparison, only *45*/141 peptides carried forward in the analysis (i.e. *45 hits*). Below are some figures to visualize the differences between these samples for considering these *hits*.

3.6.3.2 Violin Plot

Below, the violin plot visualizes the distribution of selected peptides for the analysis.

3.6.3.3 Waterfall Plot

This waterfall represents the log2 fold changes between the two groups at each peptide.

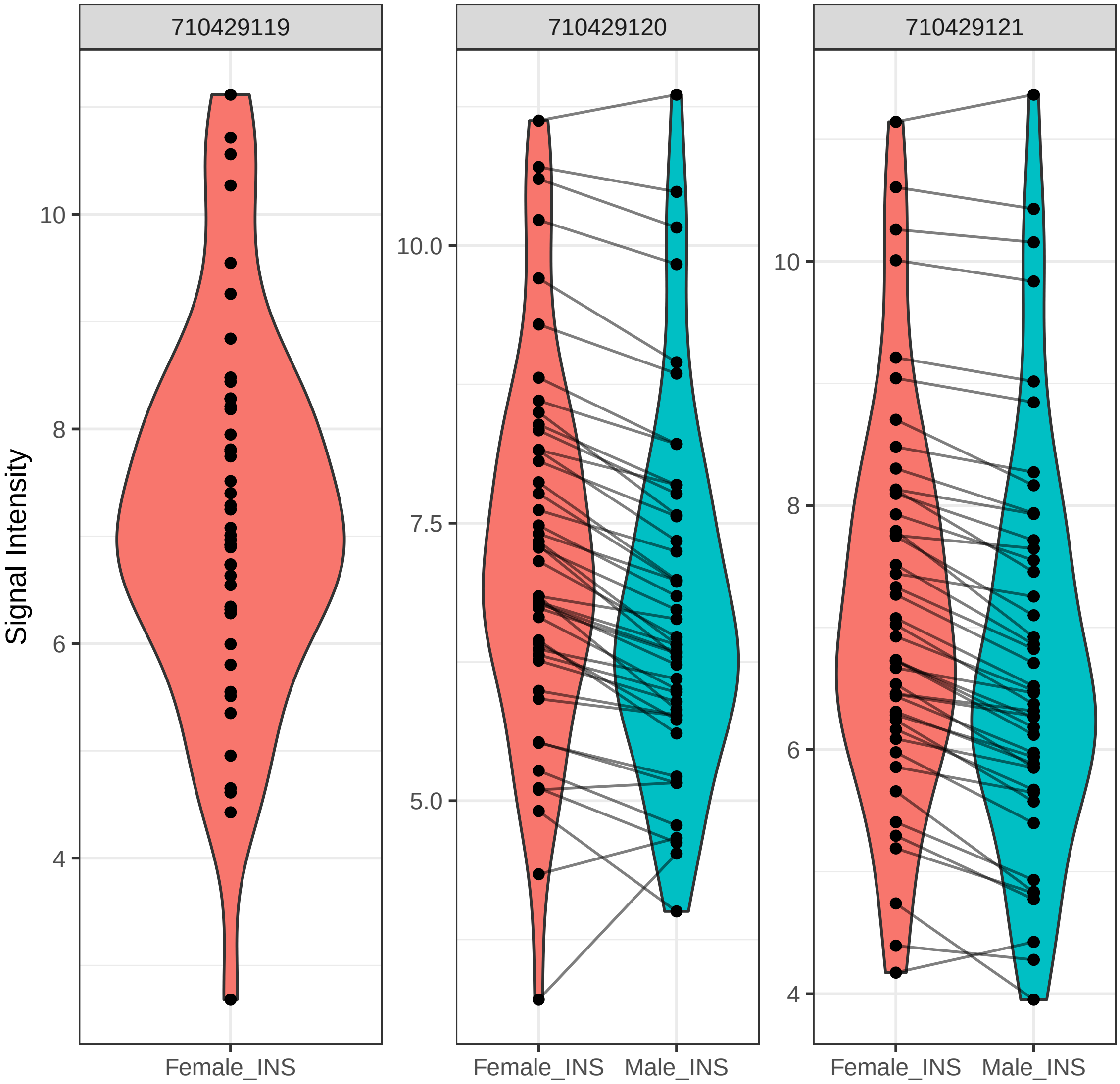


Figure 3.16: Violin plot of two groups

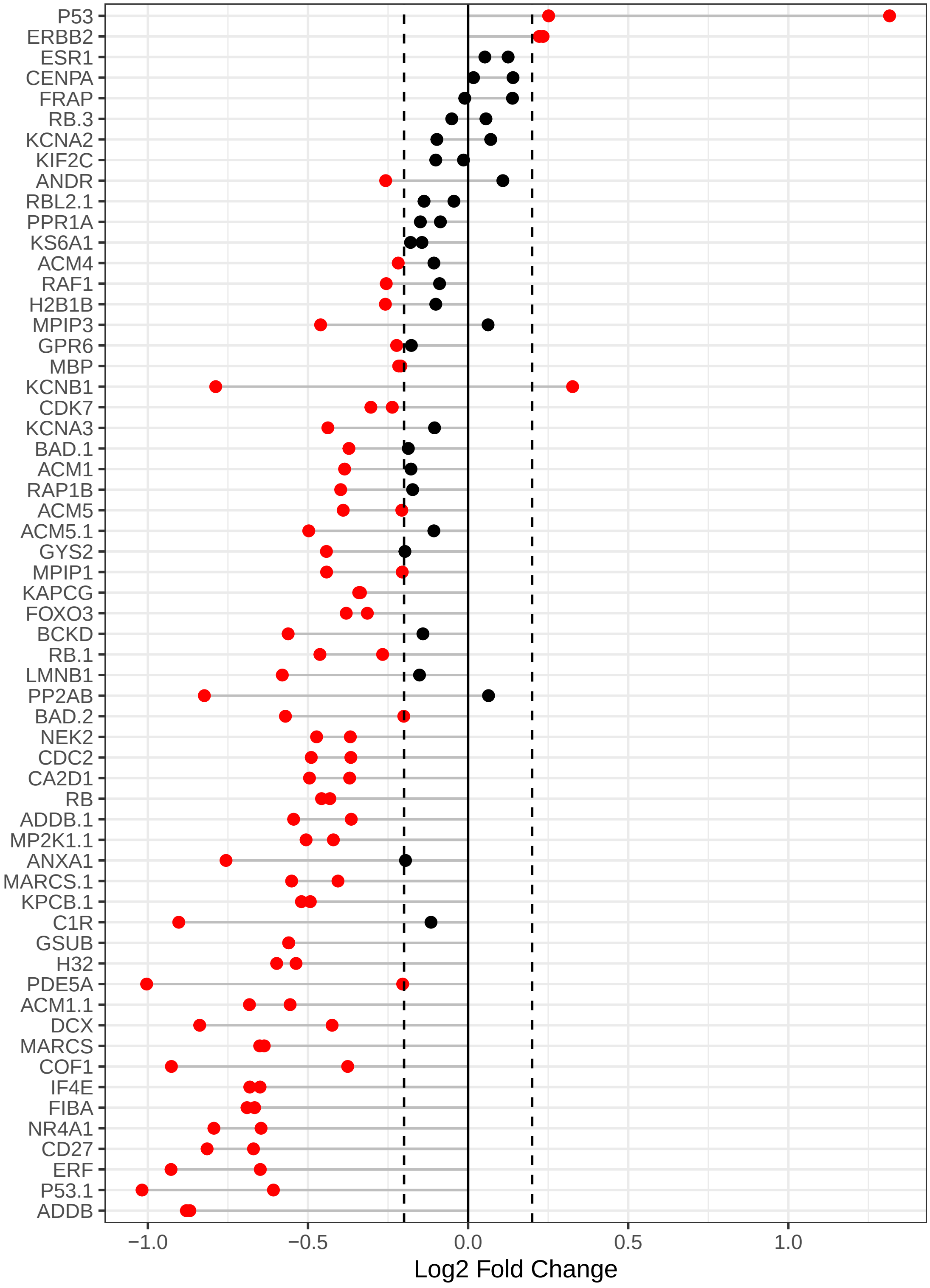


Figure 3.17: Waterfall Plot to show the distribution of change in peptide phosphorylation

3.6.3.4 Upstream Kinase Analysis

The lab carefully curated and mapped the kinases that can act and phosphorylate each peptide present on the chip. This was achieved by using multiple sources including GPS 3.0, Kinexus Phosphonet, PhosphoELM and PhosphoSite Plus. Based on that association between peptides and kinases, a random sampling analysis is performed for these hits. The basic idea of *KRSA* is: For each iteration (*2000* iterations performed in this analysis), the same number of hits are randomly selected from the total 141/or 193 peptides present on the chip. Predicted kinases are then mapped to this sample list of peptides and number of kinases are determined. The kinase count from the actual hits and random sampling is then compared to determine the significance.

Kinase	AvgZ
PDHK	-6.17500
PDK1	-6.38375
PIM	-7.00875
QIK	-7.00875
RSK	-8.37375
PHK	-10.32625
PEK	-10.44750
CDK	-10.97000
DMPK	-11.50500
PKCA	-17.29625

Method	NumberOfPeptides
meanLFC.0.15	50
meanLFC.0.2	45
meanLFC.0.3	35
meanLFC.0.4	25
710429120.0.15	49
710429120.0.2	47
710429120.0.3	38
710429120.0.4	28
710429121.0.15	42
710429121.0.2	36
710429121.0.3	29
710429121.0.4	24

3.6.3.5 Z Scores Plot

We will plot the individual and averaged Z scores using both the across and within chip analyses.

3.6.3.6 Reverse KRSA Plot

We will use the reverse KRSA plot function, to plot the log2 fold change values for all peptides mapped to kinase hits. This will help us examine the activity of the kinase

3.6.3.7 Coverage Plot

To view the coverage of kinases across the full list of peptides on the chip, we will use the coverage plot function

3.6.3.8 Ball Model Network

We will view the ball model network function, to generate a model representing the protein-protein interactions between kinases

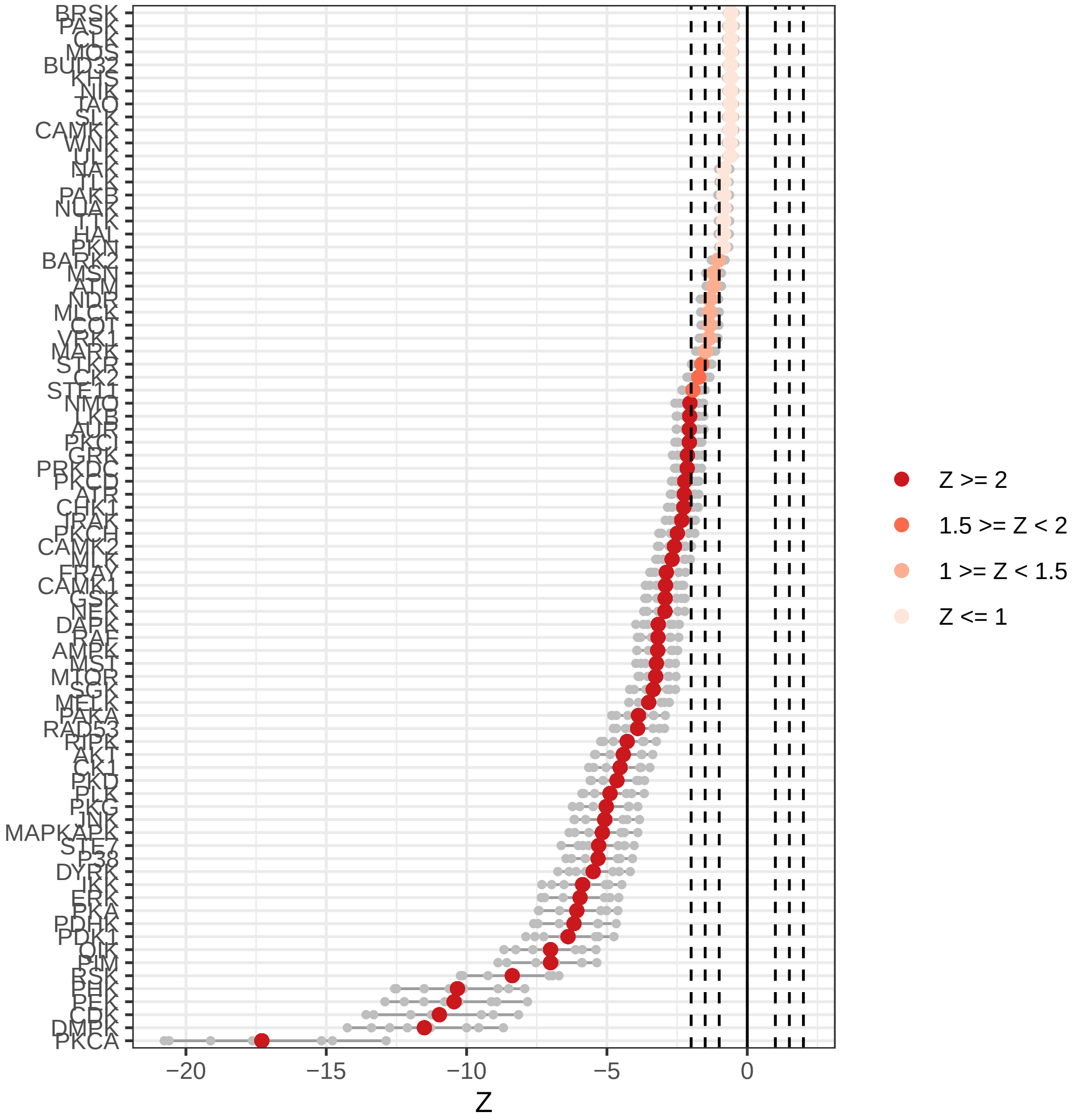


Figure 3.18: Waterfall plot of the Z Scores of each kinase family

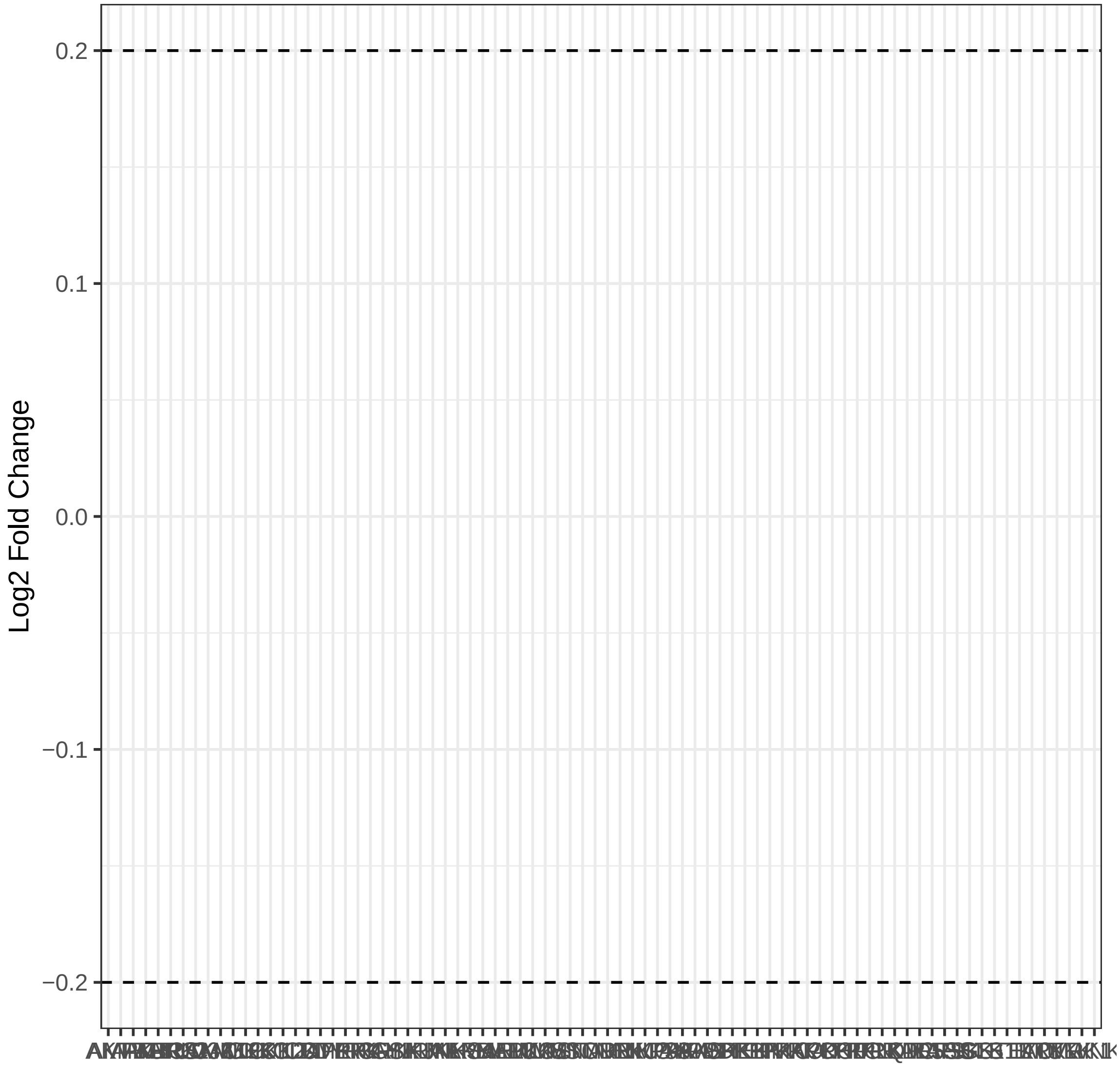


Figure 3.19: Kinase Activity summary for each kinase family based on peptide phsophorylation

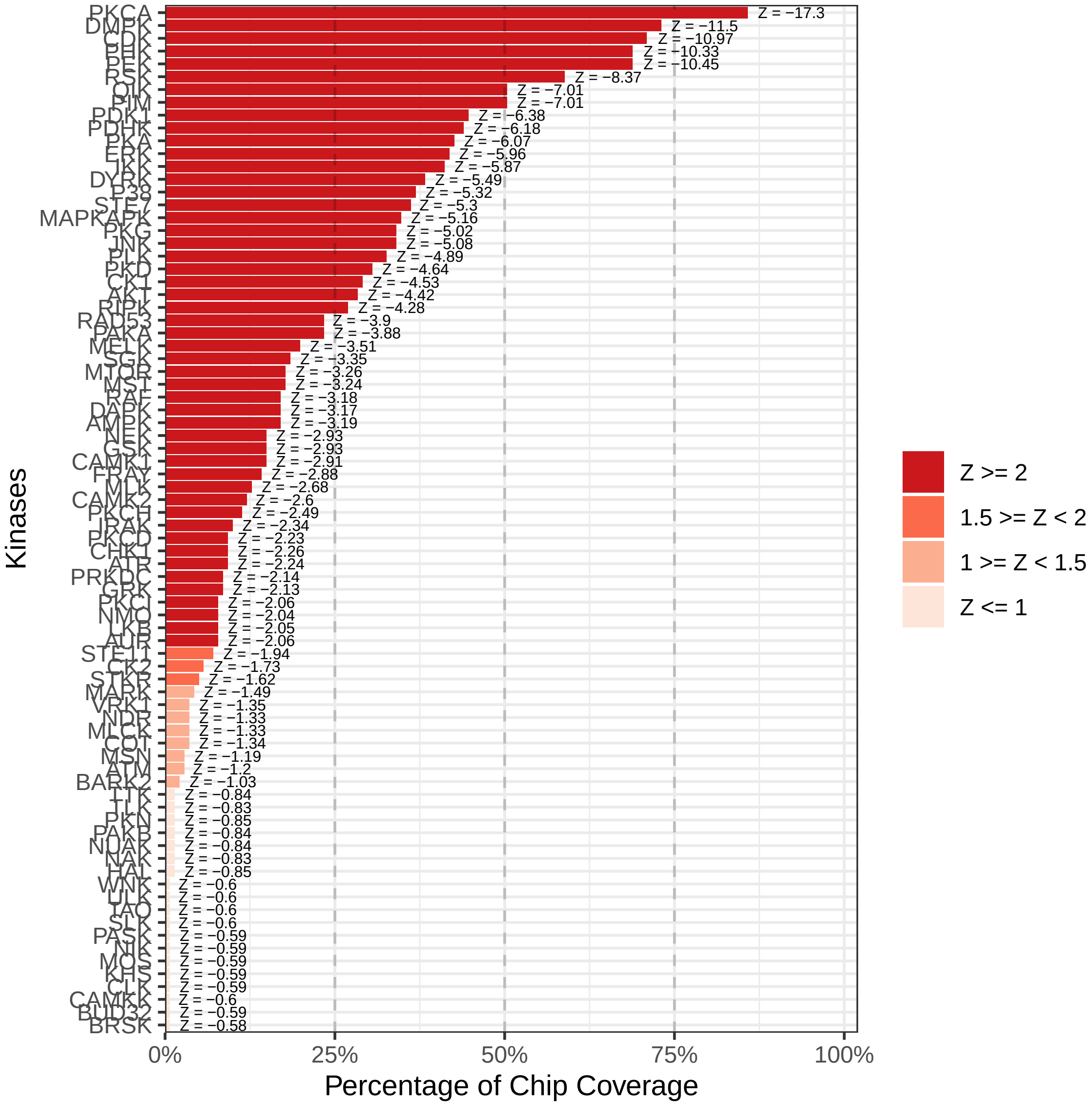
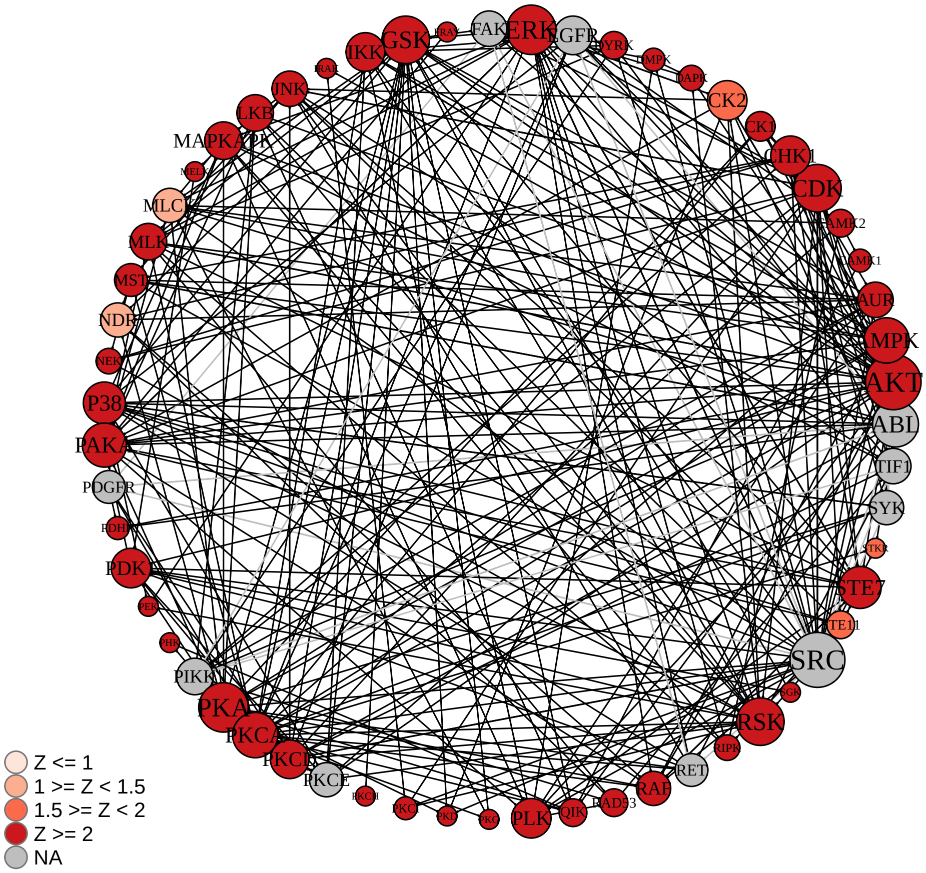
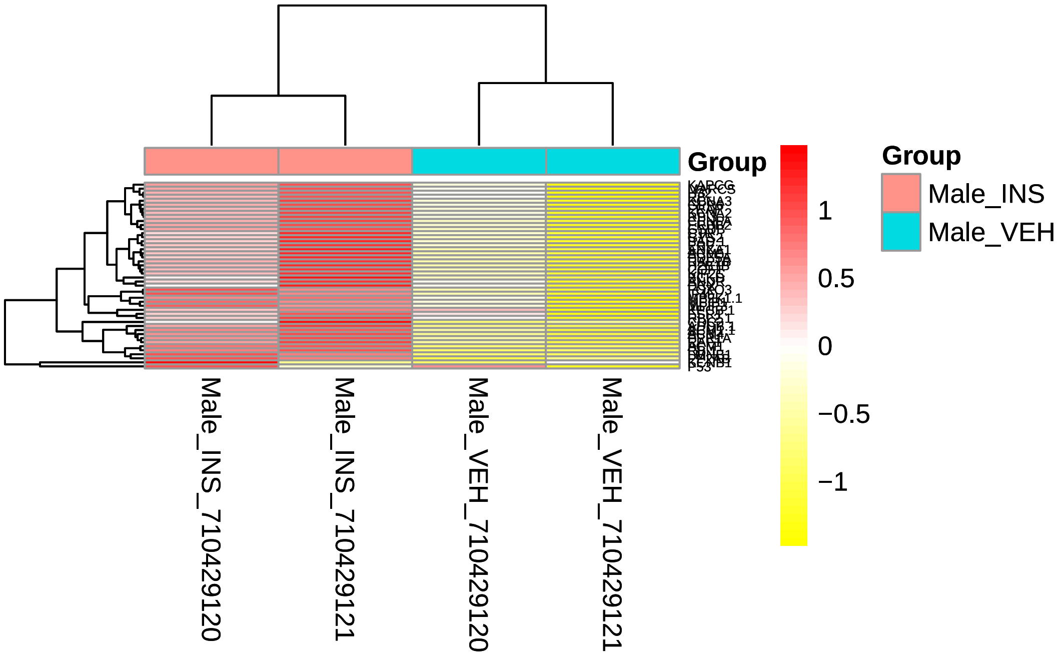


Figure 3.20: Percentge of peptides each kinase family phosphorylates


GSK
FRAY
FAK
ERK
EGFR
DYRK
DMPK
DAPK
CK2
CK1
CHK1
CDK
CAMK2
CAMK1
AUR
AMPK
AKT
ABL
TIF1
SYK
STKR
STE7
STE11
SRC
SGK
RSK
RIPK
RET
RAF
RAD53
QIK
PLK
PKG
PKD
PKCI
PKCH
PKCE
PKCD
PKCA
PKA
PIKK
PHK
PEK
PDK1
PDHK
PDGFR
PAKA
P38
NEK
NDR
MST
MLK
MLCK
MELK
MAPKAPK
LKB
JNK
IRAK
IKK
Z <= 1
1 >= Z < 1.5
1.5 >= Z < 2
Z >= 2
NA



3.6.4 Male_INS vs Male_VEH

3.6.4.1 Heatmap

After applying the *Filtering Parameters* for this group comparison, only *46*/141 peptides carried forward in the analysis (i.e. *46 hits*). Below are some figures to visualize the differences between these samples for considering these *hits*.

3.6.4.2 Violin Plot

Below, the violin plot visualizes the distribution of selected peptides for the analysis.

3.6.4.3 Waterfall Plot

This waterfall represents the log2 fold changes between the two groups at each peptide.

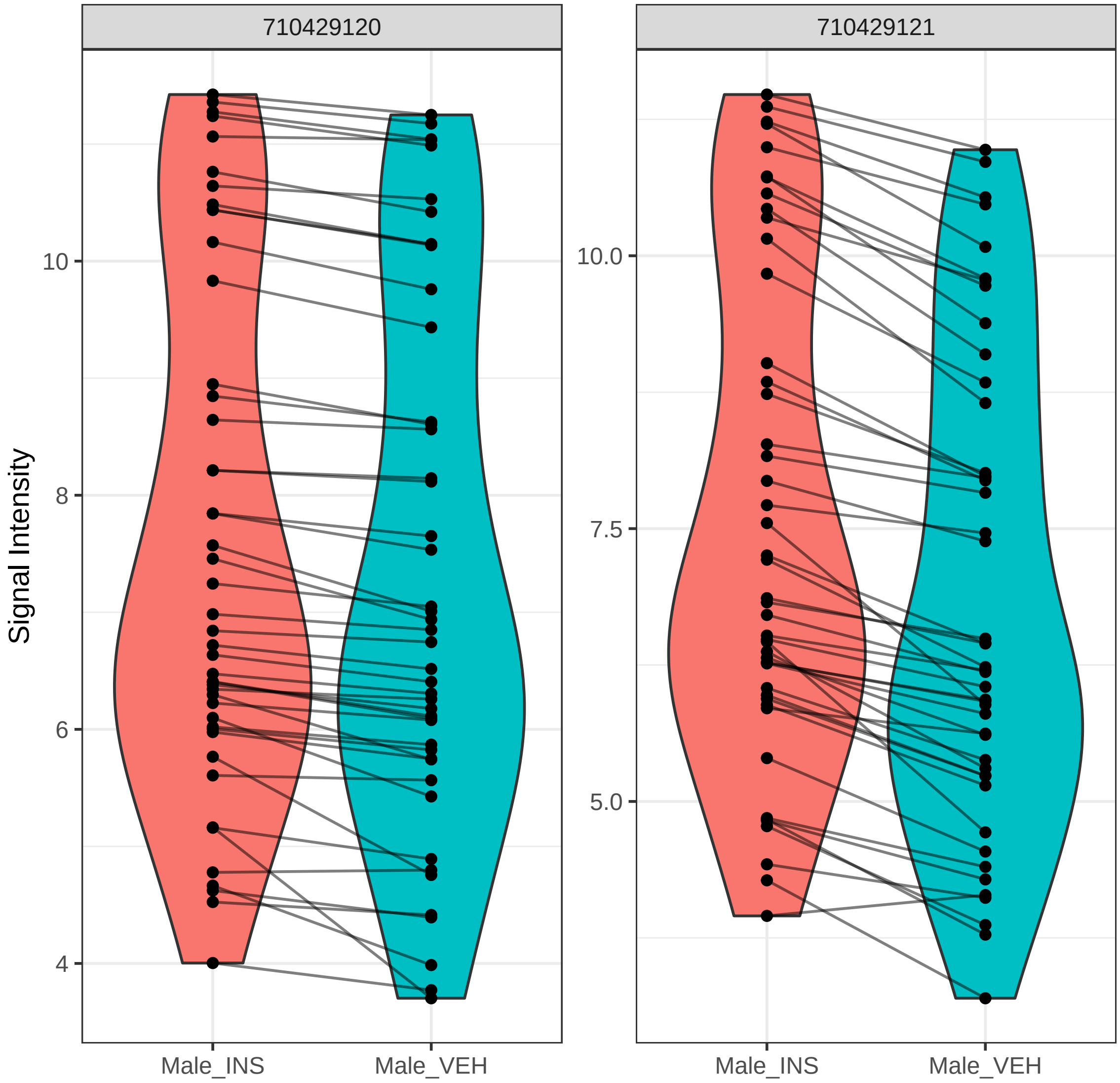


Figure 3.21: Violin plot of two groups

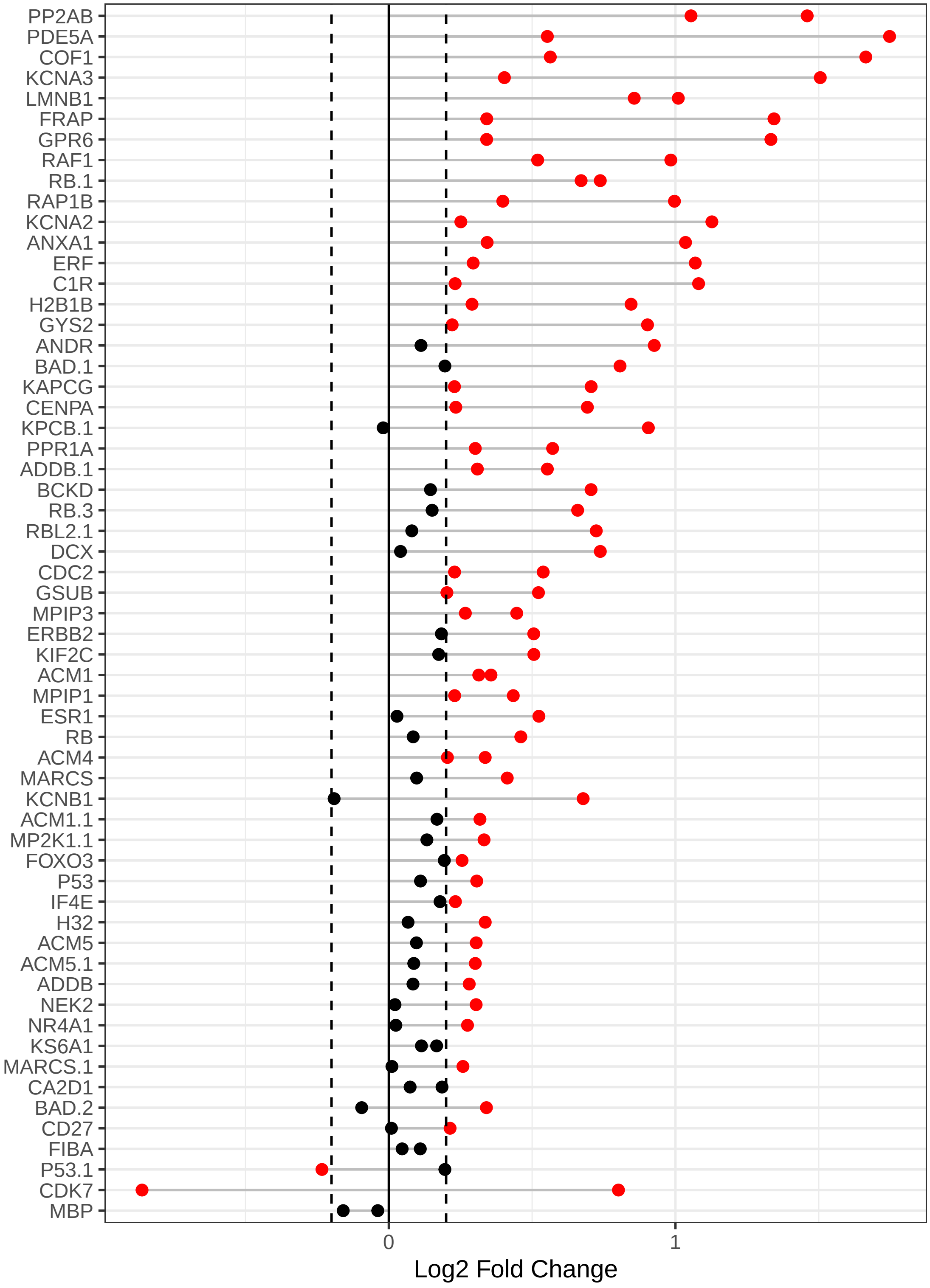


Figure 3.22: Waterfall Plot to show the distribution of change in peptide phosphorylation

3.6.4.4 Upstream Kinase Analysis

The lab carefully curated and mapped the kinases that can act and phosphorylate each peptide present on the chip. This was achieved by using multiple sources including GPS 3.0, Kinexus Phosphonet, PhosphoELM and PhosphoSite Plus. Based on that association between peptides and kinases, a random sampling analysis is performed for these hits. The basic idea of *KRSA* is: For each iteration (*2000* iterations performed in this analysis), the same number of hits are randomly selected from the total 141/or 193 peptides present on the chip. Predicted kinases are then mapped to this sample list of peptides and number of kinases are determined. The kinase count from the actual hits and random sampling is then compared to determine the significance.

Kinase	AvgZ
PDHK	-6.02875
PDK1	-6.12250
QIK	-6.86875
PIM	-6.87750
RSK	-8.09250
PEK	-10.16375
PHK	-10.21750
CDK	-10.88250
DMPK	-11.30000
PKCA	-17.06750

Method	NumberOfPeptides
meanLFC.0.15	49
meanLFC.0.2	46
meanLFC.0.3	34
meanLFC.0.4	26
710429120.0.15	37
710429120.0.2	29
710429120.0.3	16
710429120.0.4	9
710429121.0.15	57
710429121.0.2	53
710429121.0.3	47
710429121.0.4	37

3.6.4.5 Z Scores Plot

We will plot the individual and averaged Z scores using both the across and within chip analyses.

3.6.4.6 Reverse KRSA Plot

We will use the reverse KRSA plot function, to plot the log2 fold change values for all peptides mapped to kinase hits. This will help us examine the activity of the kinase

3.6.4.7 Coverage Plot

To view the coverage of kinases across the full list of peptides on the chip, we will use the coverage plot function

3.6.4.8 Ball Model Network

We will view the ball model network function, to generate a model representing the protein-protein interactions between kinases

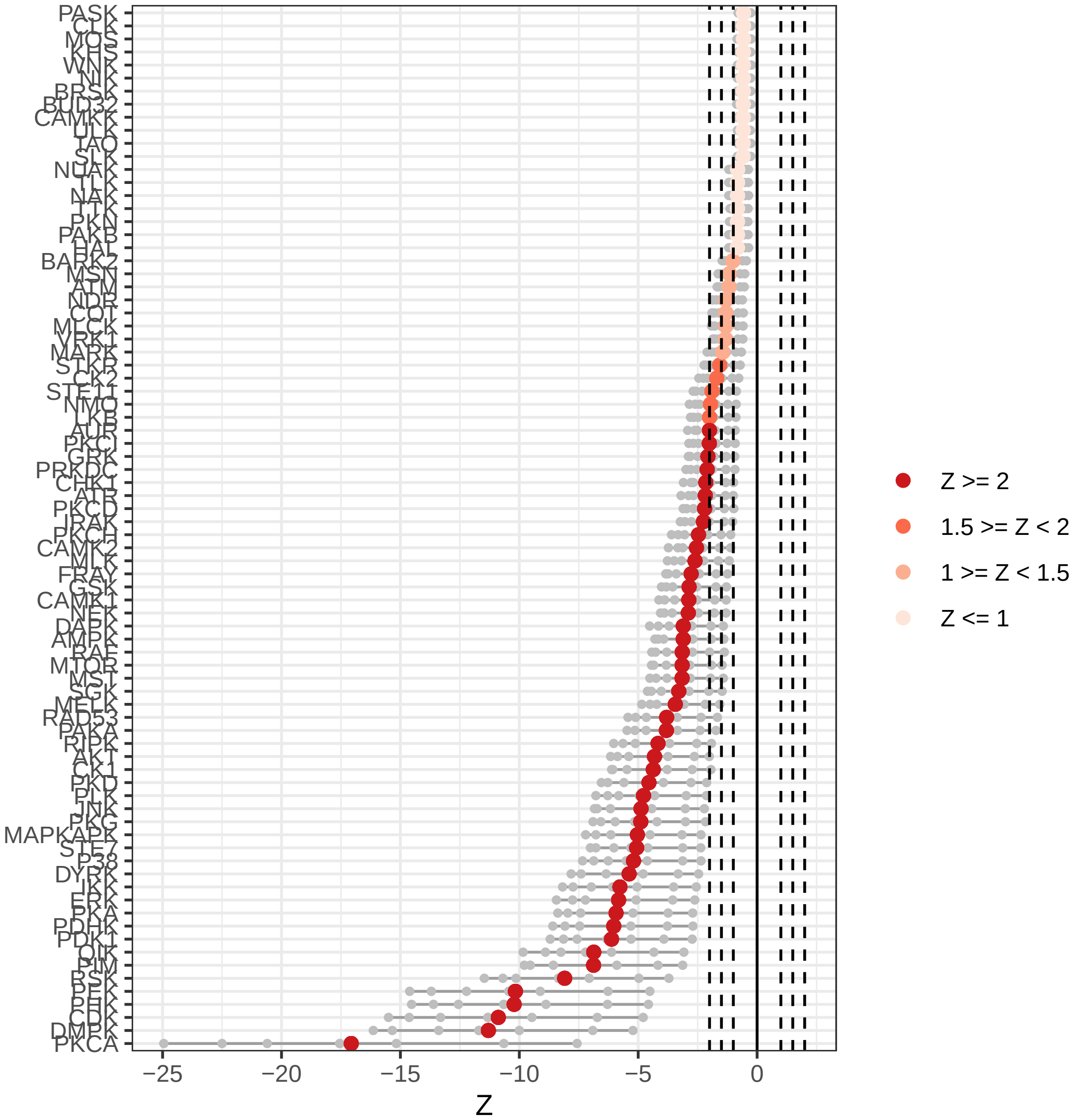


Figure 3.23: Waterfall plot of the Z Scores of each kinase family

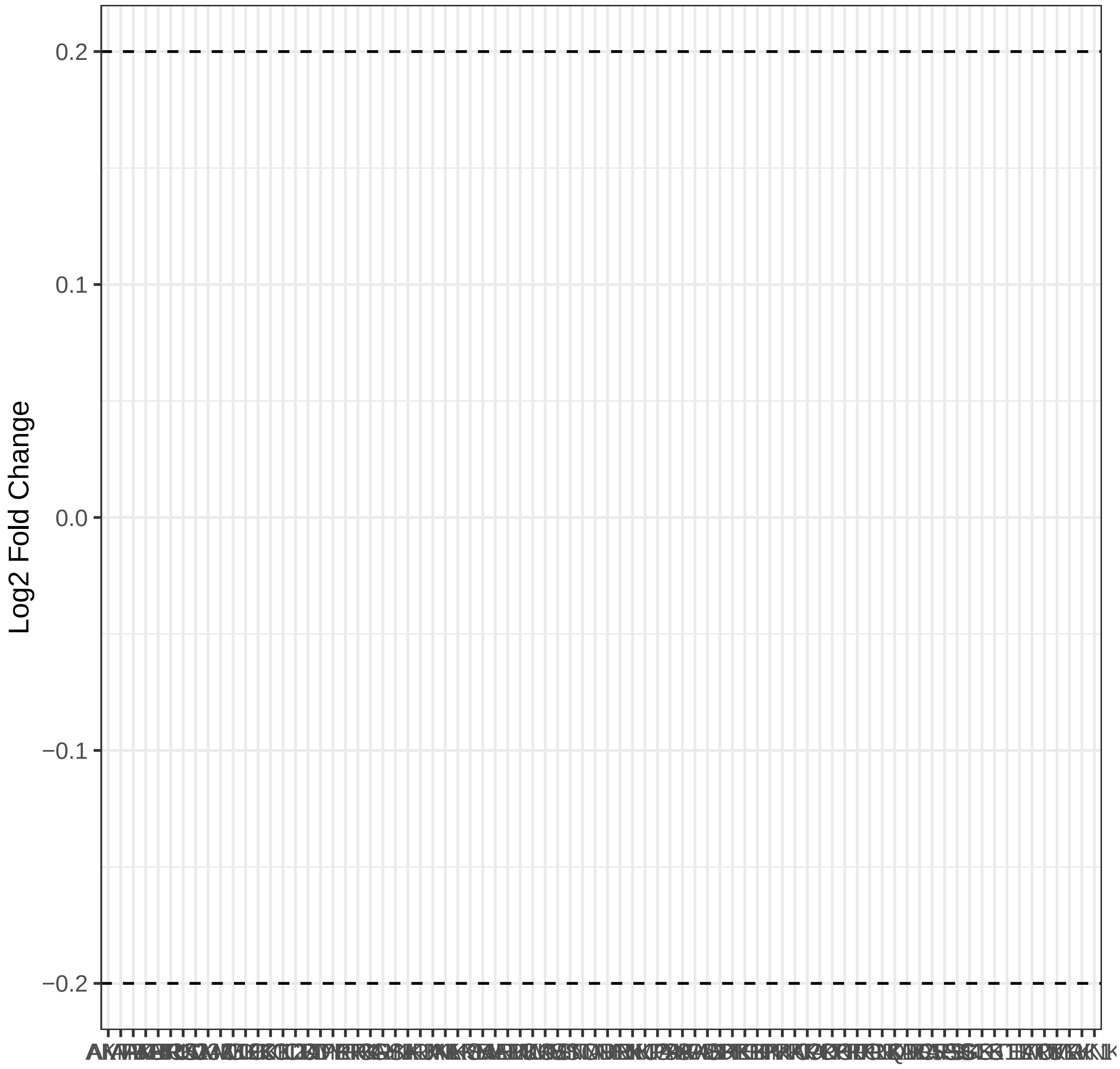


Figure 3.24: Kinase Activity summary for each kinase family based on peptide phsophorylation

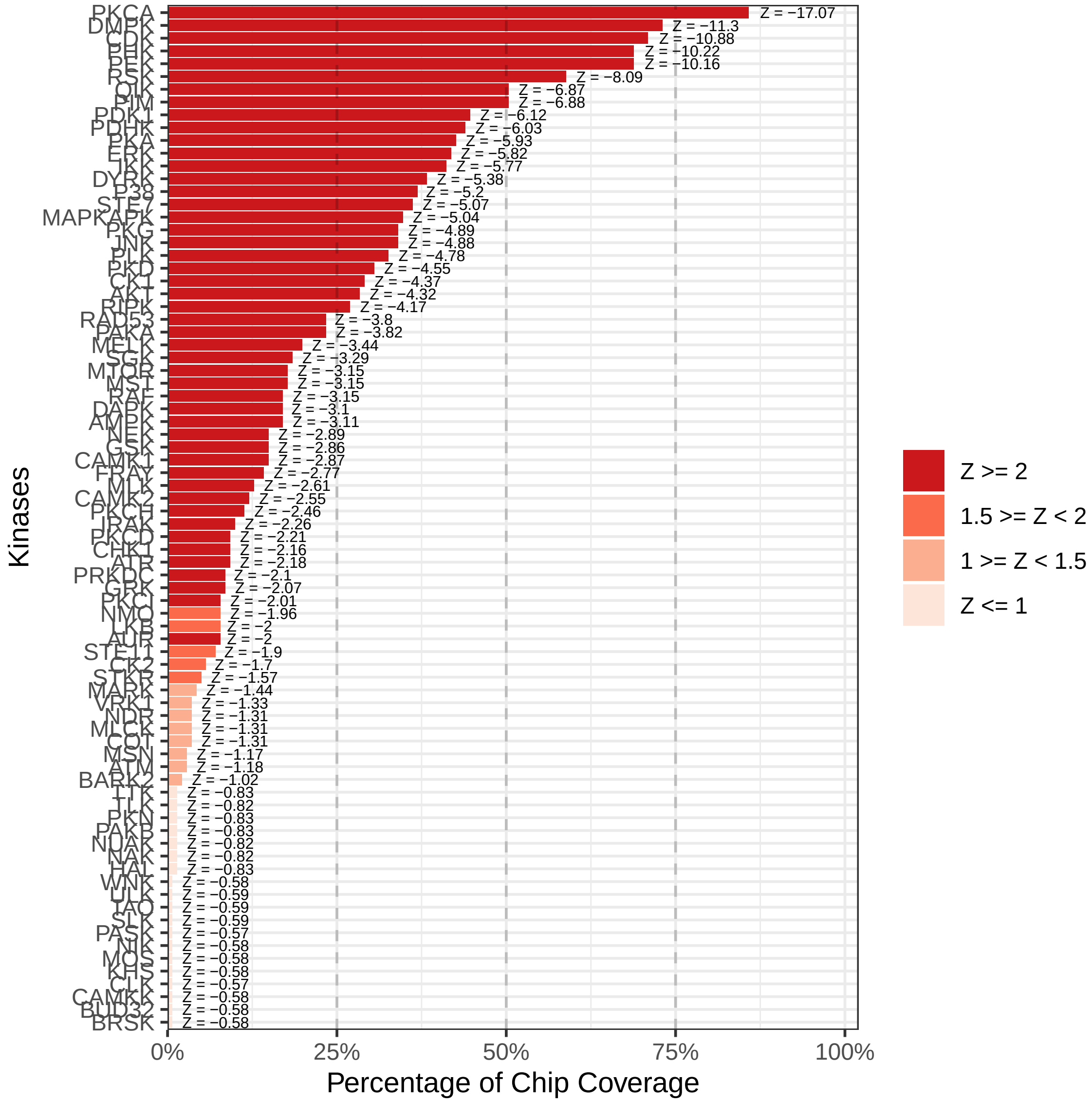
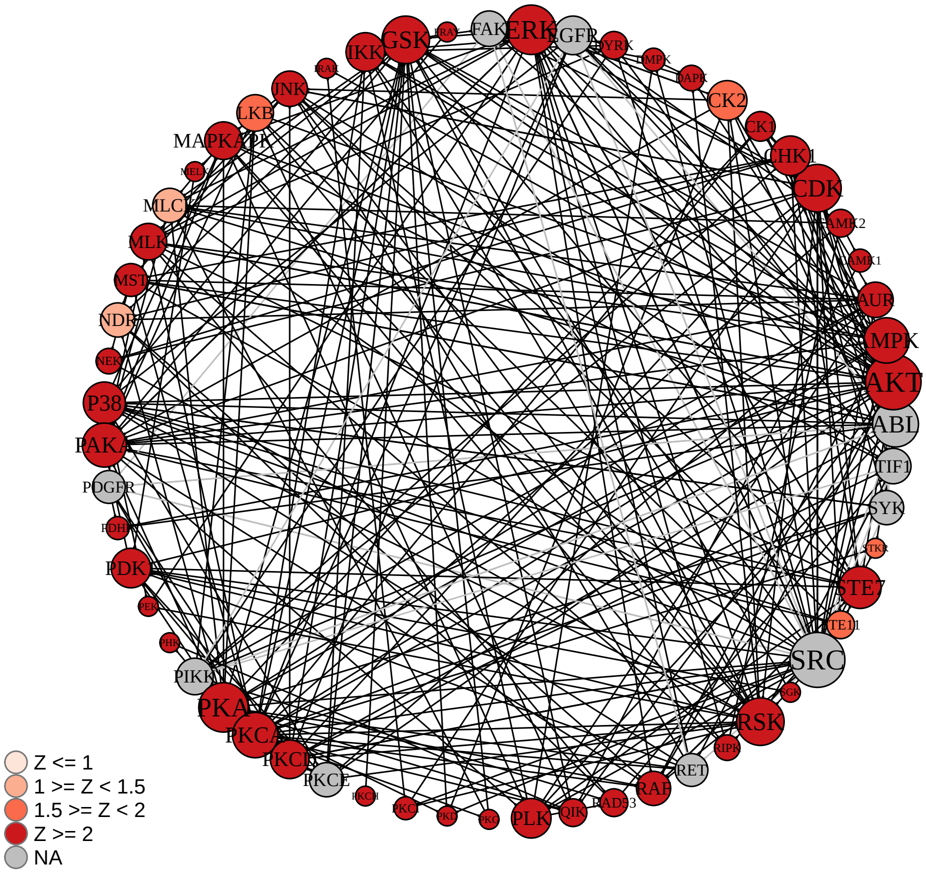


Figure 3.25: Percentge of peptides each kinase family phosphorylates


FAK
ERK
EGFR
DYRK
DMPK
DAPK
CK2
CK1
CHK1
CDK
CAMK2
CAMK1
AUR
AMPK
AKT
ABL
TIF1
SYK
STKR
STE7
STE11
SRC
SGK
RSK
RIPK
RET
RAF
RAD53
QIK
PLK
PKG
PKD
PKCI
PKCH
PKCE
PKCD
PKCA
PKA
PIKK
PHK
PEK
PDK
PDHK
PDGFR
PAKA
P38
NEK
NDR
MST
MLK
MLCK
MELK
MAPKAPK
LKB
JNK
IRAK
IKK
GSK
FRAY
Z <= 1
1 >= Z < 1.5
1.5 >= Z < 2
Z >= 2
NA

4 Session Info

```
#> - Session info ---------------------------------------------------------------
#>  setting  value
#>  version  R version 4.3.1 (2023-06-16 ucrt)
#>  os       Windows 11 x64 (build 22621)
#>  system   x86_64, mingw32
#>  ui       RTerm
#>  language (EN)
#>  collate  English_United States.utf8
#>  ctype    English_United States.utf8
#>  tz       America/New_York
#>  date     2023-08-15
#>  pandoc   3.1.1 @ C:/Program Files/RStudio/resources/app/bin/quarto/bin/tools/ (via rmarkd
#>
#> - Packages -------------------------------------------------------------------
#>  package      * version date (UTC) lib source
#>  backports      1.4.1   2021-12-13 [1] CRAN (R 4.3.0)
#>  bit            4.0.5   2022-11-15 [1] CRAN (R 4.3.1)
#>  bit64          4.0.5   2020-08-30 [1] CRAN (R 4.3.1)
#>  broom          1.0.5   2023-06-09 [1] CRAN (R 4.3.1)
#>  cachem         1.0.8   2023-05-01 [1] CRAN (R 4.3.1)
#>  callr          3.7.3   2022-11-02 [1] CRAN (R 4.3.1)
#>  cli            3.6.1   2023-03-23 [1] CRAN (R 4.3.1)
#>  codetools      0.2-19  2023-02-01 [2] CRAN (R 4.3.1)
#>  colorspace     2.1-0   2023-01-23 [1] CRAN (R 4.3.1)
#>  crayon         1.5.2   2022-09-29 [1] CRAN (R 4.3.1)
#>  devtools       2.4.5   2022-10-11 [1] CRAN (R 4.3.1)
#>  digest         0.6.33  2023-07-07 [1] CRAN (R 4.3.1)
#>  dplyr        * 1.1.2   2023-04-20 [1] CRAN (R 4.3.1)
#>  ellipsis       0.3.2   2021-04-29 [1] CRAN (R 4.3.1)
#>  EnvStats       2.8.0   2023-07-08 [1] CRAN (R 4.3.1)
#>  evaluate       0.21    2023-05-05 [1] CRAN (R 4.3.1)
#>  fansi          1.0.4   2023-01-22 [1] CRAN (R 4.3.1)
#>  farver         2.1.1   2022-07-06 [1] CRAN (R 4.3.1)
#>  fastmap        1.1.1   2023-02-24 [1] CRAN (R 4.3.1)
#>  forcats      * 1.0.0   2023-01-29 [1] CRAN (R 4.3.1)
#>  fs             1.6.3   2023-07-20 [1] CRAN (R 4.3.1)
#>  furrr        * 0.3.1   2022-08-15 [1] CRAN (R 4.3.1)
#>  future       * 1.33.0  2023-07-01 [1] CRAN (R 4.3.1)
#>  generics       0.1.3   2022-07-05 [1] CRAN (R 4.3.1)
#>  ggplot2      * 3.4.2   2023-04-03 [1] CRAN (R 4.3.1)
#>  globals        0.16.2  2022-11-21 [1] CRAN (R 4.3.0)
#>  glue           1.6.2   2022-02-24 [1] CRAN (R 4.3.1)
#>  gt           * 0.9.0   2023-03-31 [1] CRAN (R 4.3.1)
#>  gtable         0.3.3   2023-03-21 [1] CRAN (R 4.3.1)
#>  hms            1.1.3   2023-03-21 [1] CRAN (R 4.3.1)
#>  htmltools      0.5.5   2023-03-23 [1] CRAN (R 4.3.1)
#>  htmlwidgets    1.6.2   2023-03-17 [1] CRAN (R 4.3.1)
#>  httpuv         1.6.11  2023-05-11 [1] CRAN (R 4.3.1)
#>  igraph         1.5.0.1 2023-07-23 [1] CRAN (R 4.3.1)
#>  jsonlite       1.8.7   2023-06-29 [1] CRAN (R 4.3.1)
#>  knitr        * 1.43    2023-05-25 [1] CRAN (R 4.3.1)
#>  KRSA         * 1.0.0   2023-08-09 [1] Github (CogDisResLab/KRSA@0bbeca5)
#>  labeling       0.4.2   2020-10-20 [1] CRAN (R 4.3.0)
#>  later          1.3.1   2023-05-02 [1] CRAN (R 4.3.1)
#>  lattice        0.21-8  2023-04-05 [2] CRAN (R 4.3.1)
#>  lifecycle      1.0.3   2022-10-07 [1] CRAN (R 4.3.1)
#>  listenv        0.9.0   2022-12-16 [1] CRAN (R 4.3.1)
#>  lubridate    * 1.9.2   2023-02-10 [1] CRAN (R 4.3.1)
#>  magrittr       2.0.3   2022-03-30 [1] CRAN (R 4.3.1)
#>  Matrix         1.6-0   2023-07-08 [1] CRAN (R 4.3.1)
#>  memoise        2.0.1   2021-11-26 [1] CRAN (R 4.3.1)
#>  mgcv           1.9-0   2023-07-11 [1] CRAN (R 4.3.1)
#>  mime           0.12    2021-09-28 [1] CRAN (R 4.3.0)
#>  miniUI         0.1.1.1 2018-05-18 [1] CRAN (R 4.3.1)
#>  munsell        0.5.0   2018-06-12 [1] CRAN (R 4.3.1)
#>  nlme           3.1-162 2023-01-31 [2] CRAN (R 4.3.1)
#>  parallelly     1.36.0  2023-05-26 [1] CRAN (R 4.3.0)
#>  pheatmap       1.0.12  2019-01-04 [1] CRAN (R 4.3.1)
#>  pillar         1.9.0   2023-03-22 [1] CRAN (R 4.3.1)
#>  pkgbuild       1.4.2   2023-06-26 [1] CRAN (R 4.3.1)
#>  pkgconfig      2.0.3   2019-09-22 [1] CRAN (R 4.3.1)
#>  pkgload        1.3.2.1 2023-07-08 [1] CRAN (R 4.3.1)
#>  prettyunits    1.1.1   2020-01-24 [1] CRAN (R 4.3.1)
#>  processx       3.8.2   2023-06-30 [1] CRAN (R 4.3.1)
#>  profvis        0.3.8   2023-05-02 [1] CRAN (R 4.3.1)
#>  promises       1.2.0.1 2021-02-11 [1] CRAN (R 4.3.1)
#>  ps             1.7.5   2023-04-18 [1] CRAN (R 4.3.1)
#>  purrr        * 1.0.1   2023-01-10 [1] CRAN (R 4.3.1)
#>  R6             2.5.1   2021-08-19 [1] CRAN (R 4.3.1)
#>  RColorBrewer   1.1-3   2022-04-03 [1] CRAN (R 4.3.0)
#>  Rcpp           1.0.11  2023-07-06 [1] CRAN (R 4.3.1)
#>  readr        * 2.1.4   2023-02-10 [1] CRAN (R 4.3.1)
#>  remotes        2.4.2.1 2023-07-18 [1] CRAN (R 4.3.1)
#>  rlang          1.1.1   2023-04-28 [1] CRAN (R 4.3.1)
#>  rmarkdown      2.23    2023-07-01 [1] CRAN (R 4.3.1)
#>  rstudioapi     0.15.0  2023-07-07 [1] CRAN (R 4.3.1)
#>  scales         1.2.1   2022-08-20 [1] CRAN (R 4.3.1)
#>  sessioninfo    1.2.2   2021-12-06 [1] CRAN (R 4.3.1)
#>  shiny          1.7.4.1 2023-07-06 [1] CRAN (R 4.3.1)
#>  stringi        1.7.12  2023-01-11 [1] CRAN (R 4.3.0)
#>  stringr      * 1.5.0   2022-12-02 [1] CRAN (R 4.3.1)
#>  tibble       * 3.2.1   2023-03-20 [1] CRAN (R 4.3.1)
#>  tidyr        * 1.3.0   2023-01-24 [1] CRAN (R 4.3.1)
#>  tidyselect     1.2.0   2022-10-10 [1] CRAN (R 4.3.1)
#>  tidyverse    * 2.0.0   2023-02-22 [1] CRAN (R 4.3.1)
#>  timechange     0.2.0   2023-01-11 [1] CRAN (R 4.3.1)
#>  tzdb           0.4.0   2023-05-12 [1] CRAN (R 4.3.1)
#>  urlchecker     1.0.1   2021-11-30 [1] CRAN (R 4.3.1)
#>  usethis        2.2.2   2023-07-06 [1] CRAN (R 4.3.1)
#>  utf8           1.2.3   2023-01-31 [1] CRAN (R 4.3.1)
#>  vctrs          0.6.3   2023-06-14 [1] CRAN (R 4.3.1)
#>  vroom          1.6.3   2023-04-28 [1] CRAN (R 4.3.1)
#>  withr          2.5.0   2022-03-03 [1] CRAN (R 4.3.1)
#>  xfun           0.39    2023-04-20 [1] CRAN (R 4.3.1)
#>  xml2           1.3.5   2023-07-06 [1] CRAN (R 4.3.1)
#>  xtable         1.8-4   2019-04-21 [1] CRAN (R 4.3.1)
#>  yaml           2.3.7   2023-01-23 [1] CRAN (R 4.3.0)
#>
#>  [1] C:/Users/marzi/AppData/Local/R/win-library/4.3
#>  [2] C:/Program Files/R/R-4.3.1/library
#>
#> ------------------------------------------------------------------------------
```