Annex 1 | Code developed for the ClockOME analysis

ClockOME - Searching for oscillating genes in early vertebrate development

Marta Liber

Contents

1 Abstract

This R code is part of the thesis work titled: "**ClockOME:** Searching for oscillatory genes in early vertebrate development.", developed by Marta Liber © 2019/2021.

2 Packages required for this study

```
# Install the Bioconductor installer, if not already
# installed:
if (!requireNamespace("BiocManager", quietly = TRUE)) install.packages("BiocManager")

# Install the required packages:
BiocManager::install(c("devtools", "ArrayExpress", "GEOquery",
    "affy", "ggplot2", "AnnotationDbi", "arrayQualityMetrics",
    "lattice", "chicken.db", "GenomicFeatures", "annotate", "frma",
    "Oscope", "reshape", "tidyverse", "hrbrtheme", "viridis",
    "plotly", ))

# Load the installed packages

# Install the required package from GitHub repository and
# load load them:
devtools::install_github("ramiromagno/oscillation")
library(oscillation)

devtools::install_github("iduarte/frozenChicken")
library(affyChickGenomeArrayfrmavecs)
```

3 Working environment customization

3.1 Define the directories based on the tissue

- Create a folder to your project:

```
dir.create("try_clockome")
setwd("/home/mliber/try_clockome")
```

- Create a folder to download raw data:

```
dir.create("data")
data_dir <- "/home/mliber/try_clockome/data/"
```

- Create a folder to store the analysis output

```
dir.create("r_output")
output_dir <- paste0("/home/mliber/try_clockome/r_output/")
```

- Create a folder to store the analysis input files

```
dir.create("r_input")
output_dir <- paste0("/home/mliber/try_clockome/r_input/")
```

- Create a folder to store the RData and scripts

```
dir.create("rdata")
r_data <- paste0("/home/mliber/try_clockome/rdata/")
```

- Create a folder to store the data that will be analyzed (passed QC):

```
setwd("/home/mliber/try_clockome/data")
dir.create("psm")  # for PSM data
psm_data_dir <- paste0("/home/mliber/try_clockome/data/psm")

dir.create("limb")  # for limb data
limb_data_dir <- paste0("/home/mliber/try_clockome/data/limb")
```

3.2 Load functions into the global environment

1- Log2 validation of the raw microarray data (adapted from NCBI's GEO2R).

```
log_transform <- function(expression_mat) {
    qx <- as.numeric(quantile(expression_mat, c(0, 0.25, 0.5,
        0.75, 0.99, 1), na.rm = T))
    LogC <- (qx[5] > 100) || (qx[6] - qx[1] > 50 && qx[2] > 0) ||
        (qx[2] > 0 && qx[2] < 1 && qx[4] > 1 && qx[4] < 2)
    # log2 transform values if they are not log2
```

```
    if (LogC) {
        print("The RAW expression values have been log2 transformed")
        expression_mat[which(expression_mat <= 0)] <- NaN
        expression_mat <- log2(expression_mat)
    }
    return(expression_mat)
}
```

2 - Violin plot for statistical analysis of the data.

```
vio_plot <- function(df, title, yfrom, yto) {
    ggplot(df, aes(x = variable, y = value, fill = variable)) +
        geom_violin(width = 1) + geom_boxplot(width = 0.1, color = "grey",
        alpha = 0.2) + scale_fill_viridis(discrete = TRUE) +
        theme_ipsum() + theme(legend.position = "none", plot.title = element_text(size = 15),
        axis.text.x = element_text(angle = 90, hjust = 1)) +
        ggtitle(title) + xlab("") + ylim(yfrom, yto)
}
```

3 - Customized PCA plot.

```
plot_pca <- function(dataframe, PCx, PCy, title) {
    plot <- ggplot(dataframe, aes(x = dataframe[, PCx], y = dataframe[,
        PCy], color = class))
    plot <- plot + geom_point(size = 2, alpha = 0.7, show.legend = TRUE)
    plot <- plot + labs(color = "Origin") + theme_bw()
    plot <- plot + xlab(paste("PC", PCx, " (", dataframe$variance[PCx],
        "%)"))
    plot <- plot + ylab(paste("PC", PCy, " (", dataframe$variance[PCy],
        "%)"))
    plot <- plot + ggtitle(title) + theme(plot.title = element_text(size = 10))
    plot <- plot + geom_hline(yintercept = 0) + geom_vline(xintercept = 0)
    plot
}
```

4 - Creates a list of different statistics used to explore the microarray data.

```
stat_clockOME <- function(name, path, header, plot_title, classification) {
    name <- list(counts = as.matrix(read.table(path, header = header)))
    name[["data.frame"]] <- melt(as.data.frame(name$counts))
    name[["Stat_violin"]] <- vio_plot(name[["data.frame"]], plot_title,
        0, NA)
    name[["PCA_calc"]] <- prcomp(t(name$counts), center = TRUE,
        scale. = TRUE)
    name[["PCA_info"]] <- data.frame(name[["PCA_calc"]]$x, variance = as.numeric(round(100 *
        summary(name[["PCA_calc"]])$importance[2, ], digits = 2)),
        class = classification)
    name[["PCA_1"]] <- plot_pca(name[["PCA_info"]], 1, 2, plot_title)
    name[["Heatmap"]] <- heatmap(name$counts, col = viridis(256,
        1), scale = "none", Rowv = TRUE, main = plot_title)
    return(name)
}
```

5 - Functional Annotation: generation of a list of genes that are of interest.

```
my_genesOfInterest <- function(allGenesUniverse, genesOfInterest,
    allGeneNames) {
    out_geneList <- factor(as.integer(allGenesUniverse %in% genesOfInterest))
    names(out_geneList) <- allGeneNames
    return(out_geneList)
}
```

6 - Functional Annotation: analysis with *topGO* package.

```
my_topGO_analysis <- function(my_genesOfInterest,
    my_nodeSize, my_alg, my_stat) {
    # BiologicalProcesses Ontology
    go_BP_data <- new("topGOdata", ontology = "BP",
        allGenes = my_genesOfInterest, nodeSize = my_nodeSize,
        annot = annFUN.org, mapping = "org.Gg.eg.db",
        ID = "ensembl")
    n_BP_significant_genes <- as.numeric(table(go_BP_data@feasible[go_BP_data@allScores ==
        1])["TRUE"])
    go_BP_TestResults <- runTest(go_BP_data,
        algorithm = my_alg, statistic = my_stat)

    # MolecularFunction Ontology
    go_MF_data <- new("topGOdata", ontology = "MF",
        allGenes = my_genesOfInterest, nodeSize = my_nodeSize,
        annot = annFUN.org, mapping = "org.Gg.eg.db",
        ID = "ensembl")
    n_MF_significant_genes <- as.numeric(table(go_MF_data@feasible[go_MF_data@allScores ==
        1])["TRUE"])
    go_MF_TestResults <- runTest(go_MF_data,
        algorithm = my_alg, statistic = my_stat)

    # CellularComponent Ontology
    go_CC_data <- new("topGOdata", ontology = "CC",
        allGenes = my_genesOfInterest, nodeSize = my_nodeSize,
        annot = annFUN.org, mapping = "org.Gg.eg.db",
        ID = "ensembl")
    n_CC_significant_genes <- as.numeric(table(go_CC_data@feasible[go_CC_data@allScores ==
        1])["TRUE"])
    go_CC_TestResults <- runTest(go_CC_data,
        algorithm = my_alg, statistic = my_stat)

    # Gather all analysis results in a single
    # list
    results_list <- list(BP_data = go_BP_data,
        BP_TestResults = go_BP_TestResults,
        Significant_BP_genes = n_BP_significant_genes,
        MF_data = go_MF_data, MF_TestResults = go_MF_TestResults,
        Significant_MF_genes = n_MF_significant_genes,
        CC_data = go_CC_data, CC_TestResults = go_CC_TestResults,
        Significant_CC_genes = n_CC_significant_genes)
    return(results_list)
}
```

7 - Functional Annotation: generation of a table of results for each analysis.

```
my_topGO_table <- function(List_topGO_test) {
    # BiologicalProcesses
    # Ontology
    n_rows_pvalue_BP <- nrow(termStat(List_topGO_test$BP_data,
        names(score(List_topGO_test$BP_TestResults)[score(List_topGO_test$BP_TestResults) <
            0.05])))
    GO_BP_table <- GenTable(List_topGO_test$BP_data,
        List_topGO_test$BP_TestResults,
        topNodes = n_rows_pvalue_BP)
    GO_BP_table$result1 <- as.numeric(GO_BP_table$result1)
    GO_BP_table$log10pvalue <- log10(as.numeric(GO_BP_table$result1))
    GO_BP_table$Proportions <- as.numeric(GO_BP_table$Significant)/List_topGO_test$Significant_BP_genes

    # MolecularFunction
    # Ontology
    n_rows_pvalue_MF <- nrow(termStat(List_topGO_test$MF_data,
        names(score(List_topGO_test$MF_TestResults)[score(List_topGO_test$MF_TestResults) <
            0.05])))
    GO_MF_table <- GenTable(List_topGO_test$MF_data,
        List_topGO_test$MF_TestResults,
        topNodes = n_rows_pvalue_MF)
    GO_MF_table$result1 <- as.numeric(GO_MF_table$result1)
    GO_MF_table$log10pvalue <- log10(as.numeric(GO_MF_table$result1))
    GO_MF_table$Proportions <- as.numeric(GO_MF_table$Significant)/List_topGO_test$Significant_MF_genes

    # CellularComponent
    # Ontology
    n_rows_pvalue_CC <- nrow(termStat(List_topGO_test$CC_data,
        names(score(List_topGO_test$CC_TestResults)[score(List_topGO_test$CC_TestResults) <
            0.05])))
    GO_CC_table <- GenTable(List_topGO_test$CC_data,
        List_topGO_test$CC_TestResults,
        topNodes = n_rows_pvalue_CC)
    GO_CC_table$result1 <- as.numeric(GO_CC_table$result1)
    GO_CC_table$log10pvalue <- log10(as.numeric(GO_CC_table$result1))
    GO_CC_table$Proportions <- as.numeric(GO_CC_table$Significant)/List_topGO_test$Significant_CC_genes

    # Gather all analysis
    # results in a single
    # list
    results_list <- list(BP_table = GO_BP_table,
        MF_table = GO_MF_table,
        CC_table = GO_CC_table)
    return(results_list)
}
```

8 - TreeMap representation of topGO categories.

```
my_treemap <- function(GO_table, Index, vSize, title, palette) {
    treemap::treemap(GO_table, index = Index, vSize = vSize,
        type = "categorical", vColor = "Term", title = title,
        lowerbound.cex.labels = 0, bg.labels = "#CCCCCCAA", position.legend = "none",
```

```
        border.col = c("white"), border.lwds = c(3, 1), palette = palette)
}
```

9 - Lists the genes associated for each term.

```
enriched_terms <- function(my_terms, data, list_of_interest,
    ID) {
    part_1 <- function(my_terms, data, list_of_interest, ID) {
        myterms <- my_terms
        mygenes <- genesInTerm(data, myterms)
        my_enriched_genes <- lapply(mygenes, function(x) {
            list_of_interest[list_of_interest$ID %in% x, ]
        })
    }
    part_2 <- purrr::map2(mygenes, my_enriched_genes, ~list(.x,
        .y))
    return(part_2)
}
```

10 - Circular plot of p-values from enriched GO categories.

```
go_data <- ###############[,c(9,6)]
  go_data$group <- as.factor(c(rep("BP", 20), rep("MF", 20), rep("CC", 20)))
colnames(go_data) <- c("individual", "value", "group")
go_data$value <- go_data$value * 1000
data <- go_data
# data$individual <- as.character(c(1:60))
data = data %>% arrange(group, value)

# Set a number of 'empty bar' to add at the end of each group
empty_bar=3
to_add = data.frame( matrix(NA, empty_bar*nlevels(data$group), ncol(data)) )
colnames(to_add) = colnames(data)
to_add$group=rep(levels(data$group), each=empty_bar)
data=rbind(data, to_add)
data=data %>% arrange(group)
data$id=seq(1, nrow(data))

# Get the name and the y position of each label
label_data=data
number_of_bar=nrow(label_data)
angle= 90 - 360 * (label_data$id-0.5) /number_of_bar
# I substract 0.5 because the letter must have the angle of the center of the bars.
# Not extreme right(1) or extreme left (0)
label_data$hjust<-ifelse( angle < -90, 1, 0)
label_data$angle<-ifelse(angle < -90, angle+180, angle)

# prepare a data frame for base lines
base_data=data %>%
  group_by(group) %>%
  summarize(start=min(id), end=max(id) - empty_bar) %>%
  rowwise() %>%
  mutate(title=mean(c(start, end)))
```

```
# prepare a data frame for grid (scales)
grid_data = base_data
grid_data$end = grid_data$end[ c( nrow(grid_data), 1:nrow(grid_data)-1)] + 1
grid_data$start = grid_data$start - 1
grid_data=grid_data[-1,]

# Make the plot
p = ggplot(data, aes(x=as.factor(id), y=value, fill=group)) +
  geom_bar(aes(x=as.factor(id), y=value, fill=group), stat="identity", alpha=0.5) +

  # Add a val=100/75/50/25 lines. I do it at the beginning to make sure barplots are ABOVE it.
  geom_segment(data=grid_data, aes(x = end, y = 50, xend = start, yend = 50),
               colour = "grey", alpha=1, size=0.3 , inherit.aes = FALSE ) +
  geom_segment(data=grid_data, aes(x = end, y = 10, xend = start, yend = 10),
               colour = "grey", alpha=1, size=0.3 , inherit.aes = FALSE ) +
  geom_segment(data=grid_data, aes(x = end, y = 1, xend = start, yend = 1),
               colour = "grey", alpha=1, size=0.3 , inherit.aes = FALSE ) +
  geom_segment(data=grid_data, aes(x = end, y = 0.1, xend = start, yend = 0.1),
               colour = "grey", alpha=1, size=0.3 , inherit.aes = FALSE ) +
  # Add text showing the value of each 100/75/50/25 lines
  annotate("text", x = rep(max(data$id),4), y = c(50, 10, 1, 0.1),
           label = c("0.05","0.01", "0.001", "0.0001") , color="grey",
           size=3 , angle=0,fontface="bold", hjust=1) +

  geom_bar(aes(x=as.factor(id), y=value, fill=group), stat="identity", alpha=0.5) +
  ylim(-100,120) +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text = element_blank(),
    axis.title = element_blank(),
    panel.grid = element_blank(),
    plot.margin = unit(rep(-1,4), "cm")
  ) +
  coord_polar() +
  geom_text(data=label_data, aes(x=id, y=value+10, label=individual, hjust=hjust),
            color="black", fontface="bold", alpha=0.6, size=2.5,
            angle= label_data$angle, inherit.aes = FALSE ) +
  scale_color_manual(values = c("royalblue2", "red2", "seagreen3")) +
  # Add base line information
  geom_segment(data=base_data, aes(x = start, y = -5, xend = end, yend = -5),
               colour = "black", alpha=0.8, size=0.6 , inherit.aes = FALSE )  +
  geom_text(data=base_data, aes(x = title, y = -18, label=group), hjust=c(1,1,0),
            colour = "black",alpha=0.8, size=4, fontface="bold", inherit.aes = FALSE)
p
```

11 - Violin plot for each tissue (tissue profile).

```
load("/home/mliber/microarray_clockOME/both_tissues/annotation/cluster_info.RData")

tissue_info <- list(limb_counts = annotated_limb_frma, psm_counts = annotated_psm_frma,
    limb_df = melt(as.data.frame(annotated_limb_frma[, 2:15])),
    psm_df = melt(as.data.frame(annotated_psm_frma[, 2:33])))
```

```
tissue_info$limb_df$tissue <- "Limb"
tissue_info$psm_df$tissue <- "PSM"
tissue_info[["vio_limb"]] <- ggplot(tissue_info$limb_df, aes(x = tissue,
    y = value, fill = tissue)) + geom_violin(width = 1, fill = "#fec44f") +
    geom_boxplot(width = 0.1, color = "#525252", alpha = 0.7,
        fill = "white") + theme_ipsum() + theme(legend.position = "none",
    axis.text.y = element_text(size = 14), axis.text.x = element_text(angle = 90,
        hjust = 1, size = 14)) + xlab("") + ylab("")
tissue_info[["vio_psm"]] <- ggplot(tissue_info$psm_df, aes(x = tissue,
    y = value, fill = tissue)) + geom_violin(width = 1, fill = "#fcbba1") +
    geom_boxplot(width = 0.1, color = "#525252", alpha = 0.8,
        fill = "white") + theme_ipsum() + theme(legend.position = "none",
    axis.text.y = element_text(size = 14), axis.text.x = element_text(angle = 90,
        hjust = 1, size = 14)) + xlab("") + ylab("")
tissue_info[["plots"]] <- grid.arrange(tissue_info[["vio_psm"]],
    tissue_info[["vio_limb"]], nrow = 1)
```

12 - Plot the inferred trajectory of gene.

```
oscope_grid <- function(df_row, ENI_order, gene) {
    as.data.frame(df_row) %>%
        ggplot(aes(x = ENI_order, y = as.data.frame(df_row)[,
            1])) + geom_line(color = "#888a85", alpha = 0.8,
        size = 0.7) + geom_point(color = "#fcbba1", alpha = 0.8,
        size = 1.5) + cowplot::theme_cowplot() + theme(axis.text = element_text(size = 8),
        plot.title = element_text(size = 12), axis.line = element_line(colour = "#4d4d4d"),
        panel.border = element_blank(), panel.background = element_blank(),
        panel.grid.major = element_line(colour = "lightgrey",
            size = 0.1, linetype = "dashed")) + xlab("") + ylab("") +
        ggtitle(gene)
}
```

13 - Outputs the base cycle plot for a given set of points - a cluster.

```
base_cycle <- function(data, poin_col) {
    plot <- ggplot(data, aes(y = data[, 1], x = c(1:nrow(data)))) +
        geom_point(color = poin_col, alpha = 0.8, size = 4) +
        geom_line(color = "#888a85", alpha = 0.8, size = 0.7) +
        cowplot::theme_cowplot() + theme(axis.text = element_text(size = 8),
        plot.title = element_text(size = 12), axis.line = element_line(colour = "#4d4d4d"),
        panel.border = element_blank(), panel.background = element_blank(),
        panel.grid.major = element_line(colour = "lightgrey",
            size = 0.1, linetype = "dashed"))
    plot
}
```

4 The Pipeline & Script templates

4.1 Download the data

```
setwd(data_dir)
getGEOSuppFiles("GSE75798", makeDirectory = TRUE)
getAE("E-MTAB-4048", type = "raw")
getAE("E-MTAB-406", type = "raw")
```

4.2 Expand and extract the raw . CEL files

Manually expand the files and remove the .CEL files that do not correspond to chicken animal model, or that are not extracted from PSM or limb tissues.

4.3 Change the original file names to more informative new names

```
# 1-create original file names manually
original_name <- c("Gga01.CEL", "Gga02.CEL", "Gga03.CEL", "Gga04.CEL",
    "Gga05.CEL", "Gga06.CEL", "Gga07.CEL", "Gga08.CEL", "Gga09.CEL",
    "Gga10.CEL", "Gga11.CEL", "Gga12.CEL", "Gga13.CEL", "Gga14.CEL",
    "Gga15.CEL", "Gga16.CEL", "Gga17.CEL", "Gga18.CEL", "GSM1968008_OPE_PSM1dup1_Chicken.CEL",
    "GSM1968009_OPE_PSM1dup2_Chicken.CEL", "GSM1968010_OPE_PSM2dup1_Chicken.CEL",
    "GSM1968011_OPE_PSM2dup2_Chicken.CEL", "GSM1968012_OPE_PSM3dup1_Chicken.CEL",
    "GSM1968013_OPE_PSM3dup2_Chicken.CEL", "GSM1968014_OPE_PSM4dup1_Chicken.CEL",
    "GSM1968015_OPE_PSM4dup2_Chicken.CEL", "GSM1968016_OPE_PSM5dup1_Chicken.CEL",
    "GSM1968017_OPE_PSM6dup1_Chicken.CEL", "GSM1968018_OPE_PSM6dup2_Chicken.CEL",
    "GSM1968019_OPE_PSM7dup1_Chicken.CEL", "GSM1968020_OPE_PSM7dup2_Chicken.CEL",
    "GSM1968021_OPE_PSM8dup1_Chicken.CEL", "GSM1968022_OPE_PSM8dup2_Chicken.CEL.gz",
    "HH20_AL_Ant1.CEL", "HH20_AL_Ant2.CEL", "HH20_AL_Ant3.CEL",
    "HH20_AL_Ant4.CEL", "HH20_PL_Post1.CEL", "HH20_PL_Post3.CEL",
    "HH20_PL_Post4.CEL", "HH24_AL_Ant2.CEL", "HH24_AL_Ant3.CEL",
    "HH24_AL_Ant4.CEL", "HH24_AL_Ant5.CEL", "HH24_AL_Ant6.CEL",
    "HH24_PL_Post1692.CEL", "HH24_PL_Post1693.CEL", "HH24_PL_Post2.CEL",
    "HH24_PL_Post3.CEL")

# create new file names
my_ae <- paste0("ae_rPSM_", seq(1:18), ".CEL")
my_geo <- c("geo_lPSM_1", "geo_rPSM_2", "geo_lPSM_3", "geo_rPSM_4",
    "geo_lPSM_5", "geo_rPSM_6", "geo_lPSM_7", "geo_rPSM_8", "geo_lPSM_9",
    "geo_lPSM_10", "geo_rPSM_11", "geo_lPSM_12", "geo_rPSM_13",
    "geo_lPSM_14", "geo_rPSM_15")
my_geo <- paste0(my_geo, ".CEL")
my_limb <- c(paste0("ae_Alimb_", seq(1, 4), ".CEL"), paste0("ae_Plimb_",
    seq(1, 3), ".CEL"), paste0("ae_Alimb_", seq(5, 9), ".CEL"),
    paste0("ae_Plimb_", seq(4, 7), ".CEL"))

# a vector holding the characteristics of each file
classification <- c(rep("AE right PSM", 18), rep(c("Geo left PSM",
    "Geo right PSM"), 5), "Geo right PSM", "Geo left PSM", rep("AE Anterior Limb",
    4), rep("AE Posterior Limb", 3), rep("AE Anterior Limb",
```

```
    5), rep("AE Posterior Limb", 4))

# create map of old to new file names
file_name_map <- data.frame(original_name = as.character(original_name),
    new_name = as.character(c(my_ae, my_geo, my_limb)), classification = as.character(classification),
    stringsAsFactors = FALSE)

# rename files
file.rename(from = file_name_map$original_name, to = file_name_map$new_name)
```

4.4 Script for Quality control

```
# Import the data
affybatch_input <- ReadAffy(celfile.path = data_raw_dir)

# Quality Control Report
arrayQualityMetrics(expressionset = affybatch_input, outdir = paste0(output_dir,
    "QC"), force = TRUE, do.logtransform = TRUE)

# Extract and log-transformed the intensities
expres_input_raw <- exprs(affybatch_input)
expres_input_log <- log_transform(expres_input_raw)

# Boxplots of log-intensity distribution
boxplot(expres_input_log, col = rainbow(length(ncol(expres_input_log))),
    las = 3, cex.axis = 0.75, main = paste(tissue, "Log2 raw expression values",
        sep = " "))

# PCA: gives an other view of the correlations of expression
# between arrays.
pca_input <- prcomp(t(expres_input_log), center = TRUE, scale. = TRUE)
pca_input_information <- data.frame(pca_input$x, variance = as.numeric(round(100 *
    summary(pca_input)$importance[2, ], digits = 2)), class = classification)

pca_1_2_input <- plot_pca(pca_input_information, 1, 2, "input Log2 raw expression values")

pca_2_3_input <- plot_pca(pca_input_information, 2, 3, "input Log2 raw expression values")

# save the QC individual file
setwd(r_data)
save.image("qc_input.RData")
```

4.5 Script for Oscope analysis

- open and normalize the data in a new file.

```
# Load the FrozenChicken normalization vectors
data(affyChickGenomeArrayfrmavecs)

# Read the .Cel files
```

```
affybatch_input <- ReadAffy(celfile.path = data_raw_dir)

# Normalize the microarray data
eset_input_frma <- frma(affybatch_input, background = "rma",
    normalize = "quantile", summarize = "robust_weighted_average",
    target = "probeset", input.vecs = affyChickGenomeArrayfrmavecs,
    output.param = NULL, verbose = FALSE)
# Extract the expression values
expres_input_frma <- exprs(eset_input_frma)
```

- Annotation/extraction of the metadata.

```
# Select the annotation parameters
my_annotation <- AnnotationDbi::select(chicken.db, keys = (featureNames(eset_input_frma)),
    columns = c("SYMBOL", "GENENAME", "ENSEMBL"), keytype = "PROBEID")
# Remove the genes with no SYMBOL
my_annotation_sub <- subset(my_annotation, !is.na(SYMBOL))

# Merge the expression matrix with the metadata matrix
annotated_input_frma <- merge(expres_input_frma, my_annotation_sub,
    by.x = "row.names", by.y = "PROBEID", all.x = TRUE)
# Since the problem of one to many mapping persists, create a
# new unique ID (Oscope_ID)
annotated_input_frma$Oscope_ID <- paste(annotated_input_frma$Row.names,
    annotated_input_frma$GENESYMBOL, 1:nrow(annotated_input_frma),
    sep = "_")

# The values have to be exponentiated, because fRMA applies
# log2 to the expression values during the normalization
input_fnorm <- as.matrix(2^annotated_input_frma[, 2:15])
rownames(input_fnorm) <- as.vector(annotated_input_frma$Oscope_ID)
```

- Oscope analysis using *oscillation* package developed by Ramiro Magno.

```
# 1. Calculate size factors
sf_input <- oscillation::median_norm_size_factors(input_fnorm)
# 2. Normalization with EBSeq package
norm_input <- EBSeq::GetNormalizedMat(input_fnorm, sf_input)
# 3. CalcMV analogue
input_stats <- oscillation::gene_statistics(norm_input)
mv_1_input <- oscillation::filter_by_mean_count(input_stats,
    low = median(input_stats$mean))
mv_3_input <- oscillation::mean_variance_fit(mv_1_input)
mv_df_input <- data.frame(gene = as.data.frame(mv_3_input$residuals[1]),
    mean = as.data.frame(mv_3_input$residuals[2]), variance = as.data.frame(mv_3_input$residuals[3]),
    residual = as.data.frame(mv_3_input$residuals[6]))
mv_df_input <- mv_df_input[mv_df_input$.resid > 0, ]  #  genes
# 4. Rescaling between [-1;1]
rescailed_geneUse_input <- NormForSine(norm_input[mv_df_input$gene,
    ])
# 5. OscopeSine analogue
sine_input <- oscillation::paired_sine_analysis(rescailed_geneUse_input,
```

```
    parallel = TRUE, cores = 10)
# 6. Transform the tibble into matrix
sine_input_mat <- oscillation::score_matrix(as.data.frame(sine_input[,
    c(1, 2, 5)]))
shift_input_mat <- oscillation::score_matrix(as.data.frame(sine_input[,
    c(1, 2, 3)]), score = "psi")
# 7. OscopeKM analogue
km_input <- OscopeKM(list(SimiMat = sine_input_mat), maxK = 15)  #
# 8. Flagging out th clusters
flag_km_input <- FlagCluster(list(SimiMat = sine_input_mat, ShiftMat = shift_input_mat),
    KMRes = km_input, Data = rescailed_geneUse_input)
# 9. OscopeENI
ENI_input <- OscopeENI(KMRes = km_input, Data = rescailed_geneUse_input,
    NCThre = 1000)
# 10. Visualization
par(mfrow = c(2, 3))
for (i in 1:6) {
    plot(norm_input[km_input[["cluster1"]][i], ENI_input[["cluster1"]]],
        xlab = "Recovered order", ylab = "Expression", main = km_input[["cluster1"]][i])
}

# Create a matrix of original values containing only genes
# retrieved by Oscope
cluster_1_input <- annotated_input_frma %>%
    select(Oscope_ID, ENSEMBL, SYMBOL, GENENAME) %>%
    filter(Oscope_ID %in% km_input[[1]])
```

4.6 Script for Functional enrichment

```
# Generation of a list of genes that are of interest, that
# will be further explored (genes inferred by Oscope
# reasoning)
GOI_cluster <- my_genesOfInterest(annotated_cluster$Oscope_ID,
    cluster$Oscope_ID, annotated_tissue_frma$ENSEMBL)

# Functional Enrichment analysis of Gene ontologies to search
# enriched categories
cluster_enrichment <- my_topGO_analysis(GOI_cluster, my_nodeSize = 5,
    "elim", "fisher")

# Assemble BP; MF and CC metadata
GO_table_cluster <- my_topGO_table(cluster_enrichment)

sigGenes(cluster_enrichment$BP_data)
# Shows which of genes of interest were annotated i.e
# contributed to FE, do it for each database (BP, MF, CC)
```

5 Analysis of PSM tissue

5.1 Quality control

```
## Change names
setwd(data_raw_dir)

# Create original file names
original_name <- c("Gga01.CEL", "Gga02.CEL", "Gga03.CEL", "Gga04.CEL",
    "Gga05.CEL", "Gga06.CEL", "Gga07.CEL", "Gga08.CEL", "Gga09.CEL",
    "Gga10.CEL", "Gga11.CEL", "Gga12.CEL", "Gga13.CEL", "Gga14.CEL",
    "Gga15.CEL", "Gga16.CEL", "Gga17.CEL", "Gga18.CEL", "GSM1968008_OPE_PSM1dup1_Chicken.CEL",
    "GSM1968009_OPE_PSM1dup2_Chicken.CEL", "GSM1968010_OPE_PSM2dup1_Chicken.CEL",
    "GSM1968011_OPE_PSM2dup2_Chicken.CEL", "GSM1968012_OPE_PSM3dup1_Chicken.CEL",
    "GSM1968013_OPE_PSM3dup2_Chicken.CEL", "GSM1968014_OPE_PSM4dup1_Chicken.CEL",
    "GSM1968015_OPE_PSM4dup2_Chicken.CEL", "GSM1968016_OPE_PSM5dup1_Chicken.CEL",
    "GSM1968017_OPE_PSM6dup1_Chicken.CEL", "GSM1968018_OPE_PSM6dup2_Chicken.CEL",
    "GSM1968019_OPE_PSM7dup1_Chicken.CEL", "GSM1968020_OPE_PSM7dup2_Chicken.CEL",
    "GSM1968021_OPE_PSM8dup1_Chicken.CEL", "GSM1968022_OPE_PSM8dup2_Chicken.CEL.gz")

# Create new file names
my_ae <- paste0("ae_rPSM_", seq(1:18), ".CEL")
my_geo <- c("geo_lPSM_1", "geo_rPSM_2", "geo_lPSM_3", "geo_rPSM_4",
    "geo_lPSM_5", "geo_rPSM_6", "geo_lPSM_7", "geo_rPSM_8", "geo_lPSM_9",
    "geo_lPSM_10", "geo_rPSM_11", "geo_lPSM_12", "geo_rPSM_13",
    "geo_lPSM_14", "geo_rPSM_15")
my_geo <- paste0(my_geo, ".CEL")

classification <- c(rep("AE right PSM", 18), rep(c("Geo left PSM",
    "Geo right PSM"), 4), "Geo left PSM", rep(c("Geo left PSM",
    "Geo right PSM"), 3))

# Create map of old to new file names
file_name_map <- data.frame(original_name = as.character(original_name),
    new_name = as.character(c(my_ae, my_geo)), classification = as.character(classification),
    stringsAsFactors = FALSE)

# Rename files
file.rename(from = file_name_map$original_name, to = file_name_map$new_name)

# Import the psm data in an *affybatch* object
affybatch_psm <- ReadAffy(celfile.path = data_raw_dir)

# Quality Control Report
arrayQualityMetrics(expressionset = affybatch_psm, outdir = paste0(output_dir,
    "QC"), force = TRUE, do.logtransform = TRUE)
# last sample (#33) failed to QC metrics, thus need to be
# removed from the further analysis

# Remove the outlier file from data dir

# Run the Quality control again
affybatch_psm <- ReadAffy(celfile.path = data_raw_dir)
```

```
arrayQualityMetrics(expressionset = affybatch_psm, outdir = paste0(output_dir,
    "QC_2"), force = TRUE, do.logtransform = TRUE)

# Extract and log-transform the intensities
expres_psm_raw <- exprs(affybatch_psm)
expres_psm_log <- log_transform(expres_psm_raw)

# Boxplots of log-intensity distribution
boxplot(expres_psm_log, col = rainbow(length(ncol(expres_psm_log))),
    las = 3, cex.axis = 0.75, main = paste(tissue, "PSM raw expression values",
        sep = " "))

# PCA: gives an other view of the correlations of expression
# between arrays.
pca_psm <- prcomp(t(expres_psm_log), center = TRUE, scale. = TRUE)
pca_psm_information <- data.frame(pca_psm$x, variance = as.numeric(round(100 *
    summary(pca_psm)$importance[2, ], digits = 2)), class = classification[-33])

pca_1_2_psm <- plot_pca(pca_psm_information, 1, 2, "psm Log2 raw expression values")
pca_1_2_psm
pca_2_3_psm <- plot_pca(pca_psm_information, 2, 3, "psm Log2 raw expression values")
pca_2_3_psm

# save the QC individual file
setwd(r_data)
save.image("qc_psm.RData")
```

5.2 Oscope

- **Open and normalize the arrays**

```
data(affyChickGenomeArrayfrmavecs)

affybatch_psm <- ReadAffy(celfile.path = data_raw_dir)

eset_psm_frma <- frma(affybatch_psm, background = "rma", normalize = "quantile",
    summarize = "robust_weighted_average", target = "probeset",
    input.vecs = affyChickGenomeArrayfrmavecs, output.param = NULL,
    verbose = FALSE)

expres_psm_frma <- exprs(eset_psm_frma)
```

- **Genome Annotation**

```
my_annotation <- AnnotationDbi::select(chicken.db, keys = (featureNames(eset_psm_frma)),
    columns = c("SYMBOL", "GENENAME", "ENSEMBL"), keytype = "PROBEID")
my_annotation_sub <- subset(my_annotation, !is.na(SYMBOL))

annotated_psm_frma <- merge(expres_psm_frma, my_annotation_sub,
    by.x = "row.names", by.y = "PROBEID", all.x = FALSE)
annotated_psm_frma$Oscope_ID <- paste(annotated_psm_frma$Row.names,
    annotated_psm_frma$GENESYMBOL, 1:nrow(annotated_psm_frma),
```

```
    sep = "_")

# The values have to be exponentiated, because fRMA applies
# log2 to the expression values during the normalization
psm_fnorm <- as.matrix(2^annotated_psm_frma[, 2:33])
rownames(psm_fnorm) <- as.vector(annotated_psm_frma$Oscope_ID)
```

- **Trajectory Inference**

```
# 1. Calculate size factors
sf_psm <- oscillation::median_norm_size_factors(psm_fnorm)
# 2. Normalization
norm_psm <- EBSeq::GetNormalizedMat(psm_fnorm, sf_psm)
# 3. CalcMV analogue
psm_stats <- oscillation::gene_statistics(norm_psm)
mv_1_psm <- oscillation::filter_by_mean_count(psm_stats, low = quantile(psm_stats$mean,
    probs = 0.25))
# HVG extracted for downstream analysis
mv_3_psm <- oscillation::mean_variance_fit(mv_1_psm)
mv_df_psm <- data.frame(gene = as.data.frame(mv_3_psm$residuals[1]),
    mean = as.data.frame(mv_3_psm$residuals[2]), variance = as.data.frame(mv_3_psm$residuals[3]),
    residual = as.data.frame(mv_3_psm$residuals[6]))
mv_df_psm <- mv_df_psm[mv_df_psm$.resid > 0, ]  #  9527 genes

# 4. Rescaling
rescailed_geneUse_psm <- NormForSine(norm_psm[mv_df_psm$gene,
    ])
# 5. OscopeSine analogue
sine_psm <- oscillation::paired_sine_analysis(rescailed_geneUse_psm,
    parallel = TRUE, cores = 10)
# 6. Transform the tibble into matrix
sine_psm_mat <- oscillation::score_matrix(as.data.frame(sine_psm[,
    c(1, 2, 5)]))
shift_psm_mat <- oscillation::score_matrix(as.data.frame(sine_psm[,
    c(1, 2, 3)]), score = "psi")
# 7. OscopeKM analogue
km_psm <- OscopeKM(list(SimiMat = sine_psm_mat), maxK = 15)
# gene pairs above this threshold are considered:
# 1.26064932086292 max number of clusters considered:15
# optimal number of clusters:3 8. Flagging out
flag_km_psm <- FlagCluster(list(SimiMat = sine_psm_mat, ShiftMat = shift_psm_mat),
    KMRes = km_psm, Data = rescailed_geneUse_psm)
# 9. OscopeENI
ENI_psm <- OscopeENI(KMRes = km_psm, Data = rescailed_geneUse_psm,
    NCThre = 1000)
# 10. Visualization
par(mfrow = c(4, 4))  # for example
for (i in 1:16) {
    plot(norm_psm[km_psm[["cluster1"]][i], ENI_psm[["cluster1"]]],
        xlab = "Recovered order", ylab = "Expression", main = km_psm[["cluster1"]][i])
}

# Create new matrix with only relevant gene information
```

```
cluster_1_psm <- annotated_psm_frma %>%
    select(Oscope_ID, SYMBOL, GENENAME, ENSEMBL) %>%
    filter(annotated_psm_frma$Oscope_ID %in% km_psm[[1]])
cluster_1_psm_norm <- norm_psm[rownames(norm_psm) %in% km_psm[[1]],
    ENI_psm[["cluster1"]]]

cluster_2_psm <- annotated_psm_frma %>%
    select(Oscope_ID, SYMBOL, GENENAME, ENSEMBL) %>%
    filter(annotated_psm_frma$Oscope_ID %in% km_psm[[2]])
cluster_2_psm_norm <- norm_psm[rownames(norm_psm) %in% km_psm[[2]],
    ENI_psm[["cluster2"]]]
```

5.3 Functional enrichment

- ** PSM Cluster 1**

```
GOI_psm_k1 <- my_genesOfInterest(annotated_psm_frma$Oscope_ID,
    cluster_1_psm$Oscope_ID, annotated_psm_frma$ENSEMBL)
psm_k1_enrichment <- my_topGO_analysis(GOI_psm_k1, my_nodeSize = 5,
    "elim", "fisher")

GO_table_psm_k1 <- my_topGO_table(psm_k1_enrichment)
```

- ** PSM Cluster 2**

```
GOI_psm_k2 <- my_genesOfInterest(annotated_psm_frma$Oscope_ID,
    cluster_2_psm$Oscope_ID, annotated_psm_frma$ENSEMBL)
psm_k2_enrichment <- my_topGO_analysis(GOI_psm_k2, my_nodeSize = 5,
    "elim", "fisher")

GO_table_psm_k2 <- my_topGO_table(psm_k2_enrichment)
```

5.4 Descriptive Statistics and tables

```
setwd(output_dir)
# Sample ordering to plot AE vs GEO violins
initial_set_psm <- annotated_psm_frma %>%
    select(ae_rPSM_1.CEL, ae_rPSM_2.CEL, ae_rPSM_3.CEL, ae_rPSM_4.CEL,
        ae_rPSM_5.CEL, ae_rPSM_6.CEL, ae_rPSM_7.CEL, ae_rPSM_8.CEL,
        ae_rPSM_9.CEL, ae_rPSM_10.CEL, ae_rPSM_11.CEL, ae_rPSM_12.CEL,
        ae_rPSM_13.CEL, ae_rPSM_14.CEL, ae_rPSM_15.CEL, ae_rPSM_16.CEL,
        ae_rPSM_17.CEL, ae_rPSM_18.CEL, geo_lPSM_1.CEL, geo_rPSM_2.CEL,
        geo_lPSM_3.CEL, geo_rPSM_4.CEL, geo_lPSM_5.CEL, geo_rPSM_6.CEL,
        geo_lPSM_7.CEL, geo_rPSM_8.CEL, geo_lPSM_9.CEL, geo_lPSM_10.CEL,
        geo_rPSM_11.CEL, geo_lPSM_12.CEL, geo_rPSM_13.CEL, geo_lPSM_14.CEL)
rownames(initial_set_psm) <- as.vector(annotated_psm_frma$Oscope_ID)

# Initial set of genes
write.table(summary(initial_set_psm), file = paste0(output_dir,
```

```
    "/sum_psm_annot.tab"), sep = "\t", quote = FALSE)
write.table(initial_set_psm, file = paste0(output_dir, "/tab_psm_annot.tab"),
    sep = "\t", quote = FALSE)

# Putatiely oscillatory genes, filtered by 'oscillation'
filtered_genes_psm <- initial_set_psm[rownames(initial_set_psm) %in%
    mv_df_psm$gene, ]
write.table(summary(filtered_genes_psm), file = paste0(output_dir,
    "/sum_psm_oscillatory_list.tab"), sep = "\t", quote = FALSE)
write.table(filtered_genes_psm, file = paste0(output_dir, "/tab_psm_oscillatory_list.tab"),
    sep = "\t", quote = FALSE)

# Cluster 1 of 'oscillation' output
summary_k1_psm <- initial_set_psm[rownames(initial_set_psm) %in%
    km_psm[[1]], ]
write.table(summary(summary_k1_psm), file = paste0(output_dir,
    "/sum_psm_k1.tab"), sep = "\t", quote = FALSE)
write.table(summary_k1_psm, file = paste0(output_dir, "/tab_psm_k1.tab"),
    sep = "\t", quote = FALSE)

# Cluster 2 of 'oscillation' output
summary_k2_psm <- initial_set_psm[rownames(initial_set_psm) %in%
    km_psm[[2]], ]
write.table(summary(summary_k2_psm), file = paste0(output_dir,
    "/sum_psm_k2.tab"), sep = "\t", quote = FALSE)
write.table(summary_k2_psm, file = paste0(output_dir, "/tab_psm_k2.tab"),
    sep = "\t", quote = FALSE)

# Descriptive Statistics

annot_psm_ls <- stat_clockOME(annot_psm_ls, "/home/mliber/microarray_clockOME/psm/r_output/tab_psm_annot
    TRUE, plot_title = "Summarized probe intensities of psm samples",
    classification = file_name_map_psm$classification)

osci_psm_ls <- stat_clockOME(oscil_psm_ls, "/home/mliber/microarray_clockOME/psm/r_output/tab_psm_oscill
    TRUE, plot_title = "Filtered probe intensities of psm samples",
    classification = file_name_map_psm$classification)

k1_psm_ls <- stat_clockOME(k1_psm_ls, "/home/mliber/microarray_clockOME/psm/r_output/tab_psm_k1.tab",
    TRUE, plot_title = "Cluster 1 probe intensities of psm samples",
    classification = file_name_map_psm$classification)

k2_psm_ls <- stat_clockOME(k2_psm_ls, "/home/mliber/microarray_clockOME/psm/r_output/tab_psm_k2.tab",
    TRUE, plot_title = "Cluster 2 probe intensities of psm samples",
    classification = file_name_map_psm$classification)

# Save the RData
setwd(r_data)
save.image("oscillation_psm.RData")
```

6 Analysis of Limb tissue

6.1 Quality control

```
## Change names
setwd(data_raw_dir)

setwd(data_raw_dir)

# Create original file names
original_name <- c("HH20_AL_Ant1.CEL", "HH20_AL_Ant2.CEL", "HH20_AL_Ant3.CEL",
    "HH20_AL_Ant4.CEL", "HH20_PL_Post1.CEL", "HH20_PL_Post3.CEL",
    "HH20_PL_Post4.CEL", "HH24_AL_Ant2.CEL", "HH24_AL_Ant3.CEL",
    "HH24_AL_Ant4.CEL", "HH24_AL_Ant5.CEL", "HH24_AL_Ant6.CEL",
    "HH24_PL_Post1692.CEL", "HH24_PL_Post1693.CEL", "HH24_PL_Post2.CEL",
    "HH24_PL_Post3.CEL")

# Create new file names
my_limb <- c(paste0("ae_Alimb_", seq(1, 4), ".CEL"), paste0("ae_Plimb_",
    seq(1, 3), ".CEL"), paste0("ae_Alimb_", seq(5, 9), ".CEL"),
    paste0("ae_Plimb_", seq(4, 7), ".CEL"))

classification <- c(rep("AE Anterior Limb", 4), rep("AE Posterior Limb",
    3), rep("AE Anterior Limb", 5), rep("AE Posterior Limb",
    4))

# Create map of old to new file names
file_name_map <- data.frame(original_name = as.character(original_name),
    new_name = as.character(c(my_limb)), classification = as.character(classification),
    stringsAsFactors = FALSE)

# Rename files
file.rename(from = file_name_map$original_name, to = file_name_map$new_name)

# Import the psm data in an *affybatch* object
affybatch_limb <- ReadAffy(celfile.path = data_raw_dir)

# Quality Control Report
arrayQualityMetrics(expressionset = affybatch_limb, outdir = paste0(output_dir,
    "QC"), force = TRUE, do.logtransform = TRUE)
# Samples 13 and 14 failed to QC metrics, thus need to be
# removed from the further analy

# Remove the outlier file from data dir

# Run the Quality control again
expres_limb_log[, 13:14] %>%
    colnames()
affy_limb_2 <- affybatch_limb[, -c(13, 14)]

arrayQualityMetrics(expressionset = affy_limb_2, outdir = paste0(output_dir,
    "QC_2"), force = TRUE, do.logtransform = TRUE)
```

```
# Extract and log-transformed the intensities
expres_limb_raw <- exprs(affy_limb_2)
expres_limb_log <- log_transform(expres_limb_raw)

# Boxplots of log-intensity distributions
boxplot(expres_limb_log, col = rainbow(length(ncol(expres_limb_log))),
    las = 3, cex.axis = 0.75, main = paste(tissue, "Log2 raw expression values",
        sep = " "))

# PCA: gives an other view of the correlations of expression
# between arrays.
pca_limb <- prcomp(t(expres_limb_log), center = TRUE, scale. = TRUE)
pca_limb_information <- data.frame(pca_limb$x, variance = as.numeric(round(100 *
    summary(pca_limb)$importance[2, ], digits = 2)), class = classification[-c(13,
    14)])

pca_1_2_limb <- plot_pca(pca_limb_information, 1, 2, "limb Log2 raw expression values")
pca_1_2_limb

pca_2_3_limb <- plot_pca(pca_limb_information, 2, 3, "limb Log2 raw expression values")
pca_2_3_limb

# Save the RData in a specific folder
setwd(r_data)
save.image("qc_limb.RData")
```

6.2 Oscope

- **Open and normalize the arrays**

```
data(affyChickGenomeArrayfrmavecs)

affybatch_limb <- ReadAffy(celfile.path = data_raw_dir)

eset_limb_frma <- frma(affybatch_limb, background = "rma", normalize = "quantile",
    summarize = "robust_weighted_average", target = "probeset",
    input.vecs = affyChickGenomeArrayfrmavecs, output.param = NULL,
    verbose = FALSE)

expres_limb_frma <- exprs(eset_limb_frma)
```

- **Genome Annotation**

```
my_annotation <- AnnotationDbi::select(chicken.db, keys = (featureNames(eset_limb_frma)),
    columns = c("SYMBOL", "GENENAME", "ENSEMBL"), keytype = "PROBEID")
my_annotation_sub <- subset(my_annotation, !is.na(SYMBOL))

annotated_limb_frma <- merge(expres_limb_frma, my_annotation_sub,
    by.x = "row.names", by.y = "PROBEID", all.x = FALSE)
annotated_limb_frma$Oscope_ID <- paste(annotated_limb_frma$Row.names,
    annotated_limb_frma$GENESYMBOL, 1:nrow(annotated_limb_frma),
```

```
    sep = "_")

# The values have to be exponentiated, because fRMA applies
# log2 to the expression values during the normalization
limb_fnorm <- as.matrix(2^annotated_limb_frma[, 2:15])
rownames(limb_fnorm) <- as.vector(annotated_limb_frma$Oscope_ID)
```

- **Trajectory Inference**

```
# 1. size factors
sf_limb <- oscillation::median_norm_size_factors(limb_fnorm)
# 2. Normalization
norm_limb <- EBSeq::GetNormalizedMat(limb_fnorm, sf_limb)
# 3. CalcMV analogue
limb_stats <- oscillation::gene_statistics(norm_limb)
mv_1_limb <- oscillation::filter_by_mean_count(limb_stats, low = quantile(limb_stats$mean,
    probs = 0.25))
# HVG extracted for downstream analysis
mv_3_limb <- oscillation::mean_variance_fit(mv_1_limb)
mv_df_limb <- data.frame(gene = as.data.frame(mv_3_limb$residuals[1]),
    mean = as.data.frame(mv_3_limb$residuals[2]), variance = as.data.frame(mv_3_limb$residuals[3]),
    residual = as.data.frame(mv_3_limb$residuals[6]))
mv_df_limb <- mv_df_limb[mv_df_limb$.resid > 0, ]  #  10043 genes

# 4. Rescaling
rescailed_geneUse_limb <- NormForSine(norm_limb[mv_df_limb$gene,
    ])
# 5. OscopeSine analogue
sine_limb <- oscillation::paired_sine_analysis(rescailed_geneUse_limb,
    parallel = TRUE, cores = 10)
# 6. Transform the tibble into matrix
sine_limb_mat <- oscillation::score_matrix(as.data.frame(sine_limb[,
    c(1, 2, 5)]))
shift_limb_mat <- oscillation::score_matrix(as.data.frame(sine_limb[,
    c(1, 2, 3)]), score = "psi")
# 7. OscopeKM analogue
km_limb <- OscopeKM(list(SimiMat = sine_limb_mat), maxK = 15)
# 8. Flagging out
flag_km_limb <- FlagCluster(list(SimiMat = sine_limb_mat, ShiftMat = shift_limb_mat),
    KMRes = km_limb, Data = rescailed_geneUse_limb)
# 9. OscopeENI
ENI_limb <- OscopeENI(KMRes = km_limb, Data = rescailed_geneUse_limb,
    NCThre = 1000)
# 9. Visualization
par(mfrow = c(2, 2))
for (i in 17:20) {
    plot(norm_limb[km_limb[["cluster1"]][i], ENI_limb[["cluster1"]]],
        xlab = "Recovered order", ylab = "Expression", main = cluster_1_limb[i,
            "SYMBOL"], type = "b", col = "#fec44f")
}

# Extract the annotation information from the oscope output
cluster_1_limb <- annotated_limb_frma %>%
```

```
    select(Oscope_ID, SYMBOL, GENENAME, ENSEMBL) %>%
    filter(annotated_limb_frma$Oscope_ID %in% km_limb[[1]])
cluster_1_limb_norm <- norm_limb[rownames(norm_limb) %in% km_limb[[1]],
    ENI_limb[["cluster1"]]]
```

6.3 Functional enrichment

```
# PSM K1
setwd("/home/mliber/microarray_clockOME/limb/r_output/topGO/")
GOI_limb <- my_genesOfInterest(annotated_limb_frma$Oscope_ID,
    cluster_1_limb$Oscope_ID, annotated_limb_frma$ENSEMBL)
limb_enrichment <- my_topGO_analysis(GOI_limb, my_nodeSize = 5,
    "elim", "fisher")

GO_table_limb <- my_topGO_table(limb_enrichment)
```

6.4 Descriptive Statistics and tables

```
setwd(output_dir)
# Table of descriptive statistics
setwd(output_dir)
initial_set_limb <- annotated_limb_frma[, 2:15]  #frozenchicken normalized + log2 transformed
rownames(initial_set_limb) <- as.vector(annotated_limb_frma$Oscope_ID)

# Initial set of genes
write.table(summary(initial_set_limb), file = paste0(output_dir,
    "/sum_limb_annot.tab"), sep = "\t", quote = FALSE)
write.table(initial_set_limb, file = paste0(output_dir, "/tab_limb_annot.tab"),
    sep = "\t", quote = FALSE)

# Putatively oscillatory filtered by 'oscillation'
filtered_genes_limb <- initial_set_limb[rownames(initial_set_limb) %in%
    mv_df_limb$gene, ]
write.table(summary(filtered_genes_limb), file = paste0(output_dir,
    "/sum_limb_oscillatory_list.tab"), sep = "\t", quote = FALSE)
write.table(filtered_genes_limb, file = paste0(output_dir, "/tab_limb_oscillatory_list.tab"),
    sep = "\t", quote = FALSE)

# Cluster 1 of 'oscillation' output
summary_k1_limb <- initial_set_limb[rownames(initial_set_limb) %in%
    km_limb[[1]], ]
write.table(summary(summary_k1_limb), file = paste0(output_dir,
    "/sum_limb_k1.tab"), sep = "\t", quote = FALSE)
write.table(summary_k1_limb, file = paste0(output_dir, "/tab_limb_k1.tab"),
    sep = "\t", quote = FALSE)

# Cluster ENSEMBL IDs
write.table(cluster_1_limb, file = "K1_limb.tab")
```

```
# Descriptive Statistics
annot_limb_ls <- stat_clockOME(annot_limb_ls, "/home/mliber/microarray_clockOME/limb/r_output/tab_limb_a
    TRUE, plot_title = "Summarized probe intensities of limb samples",
    classification = file_name_map_limb$classification)

osci_limb_ls <- stat_clockOME(oscil_limb_ls, "/home/mliber/microarray_clockOME/limb/r_output/tab_limb_os
    TRUE, plot_title = "Filtered probe intensities of limb samples",
    classification = file_name_map_limb$classification)

k1_limb_ls <- stat_clockOME(k1_limb_ls, "/home/mliber/microarray_clockOME/limb/r_output/tab_limb_k1.tab'
    TRUE, plot_title = "Cluster 1 probe intensities of limb samples",
    classification = file_name_map_limb$classification)
# Save the Rdata
setwd(r_data)
save.image("oscillation_limb.RData")
```

7 Crossing the list of genes between different studies

- **F.1 - ClockOME list vs Matsuda (2017) | Kroll (2011) lists**

```
# 1. Extract the gene symbols from the supplementary list, from both studies:
# Load the kroll list o f 636 gene symbols for chicken
krol <- read.table(file = "/home/mliber/microarray_clockOME/both_tissues/krol_list.txt")
# Load the Matsuda list o f 220 gene symbols for human iPSC
matsuda <- read.table(file = "/home/mliber/microarray_clockOME/both_tissues/matsuda_list.txt")

# examine the crossing for ClockOME limb vs Krol
cluster_1_limb[grep("TRUE", cluster_1_limb$SYMBOL %in% as.vector(krol$V1)), ]  # 12
# ClockOME Limb vs Matsuda
cluster_1_limb[grep("TRUE", cluster_1_limb$SYMBOL %in% as.vector(matsuda$V1)), ]  # 4

# examine the crossing for ClockOME PSM cluster PSM K1 vs Krol
cluster_1_psm[grep("TRUE", cluster_1_psm$SYMBOL %in% as.vector(krol$V1)), ]  # 1
# PSM K1 vs Matsuda
cluster_1_psm[grep("TRUE", cluster_1_psm$SYMBOL %in% as.vector(matsuda$V1)), ]  # 4
# However FGFR1 x2 and RHOU x2 probes PSM K2 vs
cluster_2_psm[grep("TRUE", cluster_2_psm$SYMBOL %in% as.vector(krol$V1)), ]  # 1
# PSM K2 vs Matsuda
cluster_2_psm[grep("TRUE", cluster_2_psm$SYMBOL %in% as.vector(matsuda$V1)), ]  # 0
```

- **F.1 - ClockOME Limb vs PSM K1 | K2**

```
# between both PSM clusters
as.vector(cluster_1_psm$SYMBOL) %in% as.vector(cluster_2_psm$SYMBOL)
# not possible

# between both PSM K1 vs Limb clusters
cluster_1_limb[grep("TRUE", as.vector(cluster_1_psm$SYMBOL) %in%
    as.vector(cluster_1_limb$SYMBOL)), ]  # 3
# However FSTL4 x2 probes

# between both PSM K2 vs Limb clusters
cluster_1_limb[grep("TRUE", as.vector(cluster_2_psm$SYMBOL) %in%
    as.vector(cluster_1_limb$SYMBOL)), ]  # 2
```

Annex 2 | Quality Control of the microarray datasets

Annex 2.1 - Quality Control of the Limb microarray datasets

arrayQualityMetrics report for affybatch_limb

- Array metadata and outlier detection overview

	array	sampleNames	*1	*2	*3	*4	*5	*6	sample	ScanDate
☐	1	HH20_AL_Ant1.CEL							1	03/15/07 16:20:06
☐	2	HH20_AL_Ant2.CEL							2	03/15/07 16:29:05
☐	3	HH20_AL_Ant3.CEL							3	03/15/07 16:37:56
☐	4	HH20_AL_Ant4.CEL							4	03/15/07 16:46:36
☐	5	HH20_PL_Post1.CEL							5	03/15/07 16:55:43
☐	6	HH20_PL_Post3.CEL							6	03/15/07 17:04:17
☐	7	HH20_PL_Post4.CEL							7	03/15/07 17:17:21
☐	8	HH24_AL_Ant2.CEL							8	12/06/05 12:41:38
☐	9	HH24_AL_Ant3.CEL							9	12/06/05 12:52:14
☐	10	HH24_AL_Ant4.CEL							10	12/06/05 13:02:50
☐	11	HH24_AL_Ant5.CEL							11	12/06/05 13:13:22
☐	12	HH24_AL_Ant6.CEL							12	12/06/05 13:30:09
☑	13	HH24_PL_Post1692.CEL				x			13	07/07/09 06:24:27
☑	14	HH24_PL_Post1693.CEL				x			14	07/07/09 06:15:30
☐	15	HH24_PL_Post2.CEL							15	12/06/05 13:41:12
☐	16	HH24_PL_Post3.CEL							16	12/06/05 13:51:39

The columns named *1, *2, ... indicate the calls from the different outlier detection methods:

1. outlier detection by Distances between arrays
2. outlier detection by Boxplots
3. outlier detection by Relative Log Expression (RLE)
4. outlier detection by Normalized Unscaled Standard Error (NUSE)
5. outlier detection by MA plots
6. outlier detection by Spatial distribution of M

The outlier detection criteria are explained below in the respective sections. Arrays that were called outliers by at least one criterion are marked by checkbox selection in this table, and are indicated by highlighted lines or points in some of the plots below. By clicking the checkboxes in the table, or on the corresponding points/lines in the plots, you can modify the selection. To reset the selection, reload the HTML page in your browser.

At the scope covered by this software, outlier detection is a poorly defined question, and there is no 'right' or 'wrong' answer. These are hints which are intended to be followed up manually. If you want to automate outlier detection, you need to limit the scope to a particular platform and experimental design, and then choose and calibrate the metrics used.

Section 1: Between array comparison

- Figure 1: Distances between arrays.

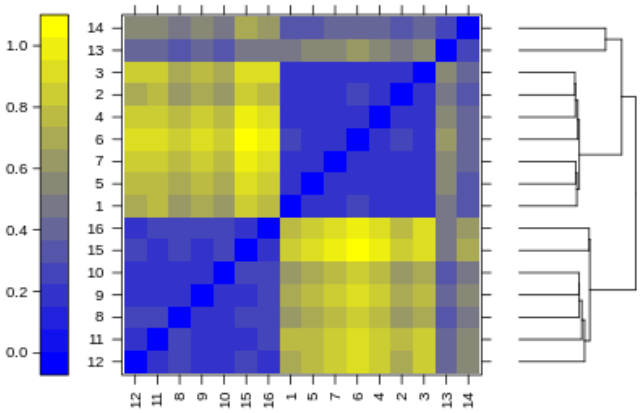


Figure 1 (PDF file) shows a false color heatmap of the distances between arrays. The color scale is chosen to cover the range of distances encountered in the dataset. Patterns in this plot can indicate clustering of the arrays either because of intended biological or unintended experimental factors (batch effects). The distance d_{ab} between two arrays *a* and *b* is computed as the mean absolute difference (L_1-distance) between the data of the arrays (using the data from all probes without filtering). In formula, d_{ab} = mean | M_{ai} - M_{bi} |, where M_{ai} is the value of the *i*-th probe on the *a*-th array. Outlier detection was performed by looking for arrays for which the sum of the distances to all other arrays, S_a = Σ_b d_{ab} was exceptionally large. No such arrays were detected.

- Figure 2: Outlier detection for Distances between arrays.

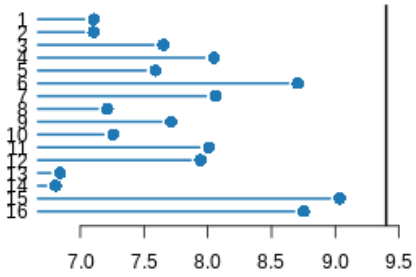


Figure 2 (PDF file) shows a bar chart of the sum of distances to other arrays S_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 9.4 was determined, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

- Figure 3: Principal Component Analysis.

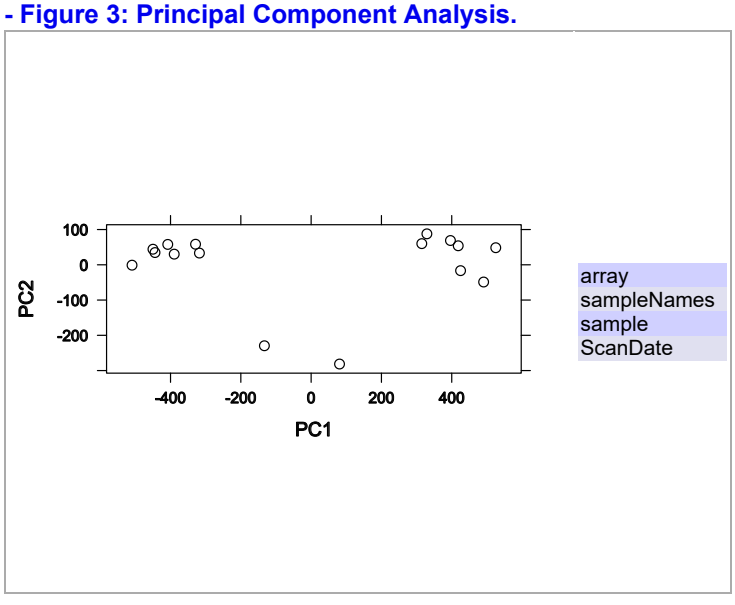


Figure 3 (PDF file) shows a scatterplot of the arrays along the first two principal components. You can use this plot to explore if the arrays cluster, and whether this is according to an intended experimental factor (you can indicate such a factor by color using the 'intgroup' argument), or according to unintended causes such as batch effects. Move the mouse over the points to see the sample names.
Principal component analysis is a dimension reduction and visualisation technique that is here used to project the multivariate data vector of each array into a two-dimensional plot, such that the spatial arrangement of the points in the plot reflects the overall data (dis)similarity between the arrays.

Section 2: Array intensity distributions

- Figure 4: Boxplots.

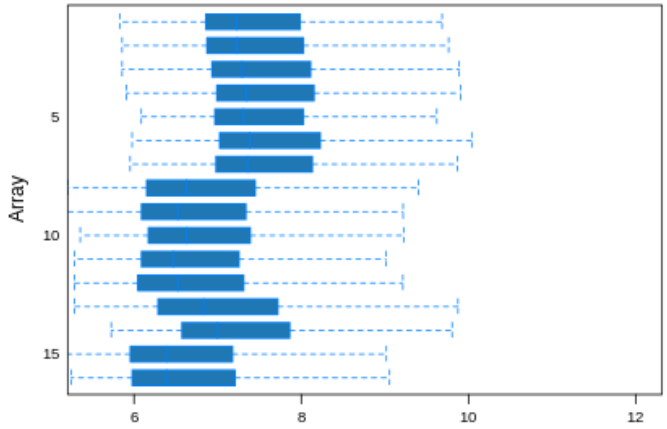


Figure 4 (PDF file) shows boxplots representing summaries of the signal intensity distributions of the arrays. Each box corresponds to one array. Typically, one expects the boxes to have similar positions and widths. If the distribution of an array is very different from the others, this may indicate an experimental problem. Outlier detection was performed by computing the Kolmogorov-Smirnov statistic K_a between each array's distribution and the distribution of the pooled data.

- Figure 5: Outlier detection for Boxplots.

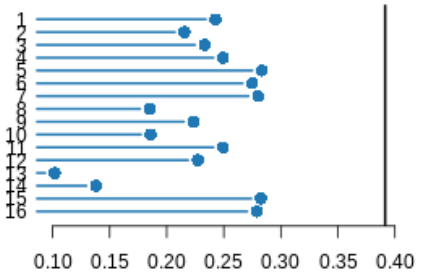


Figure 5 (PDF file) shows a bar chart of the Kolmogorov-Smirnov statistic K_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 0.391 was determined, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

- Figure 6: Density plots.

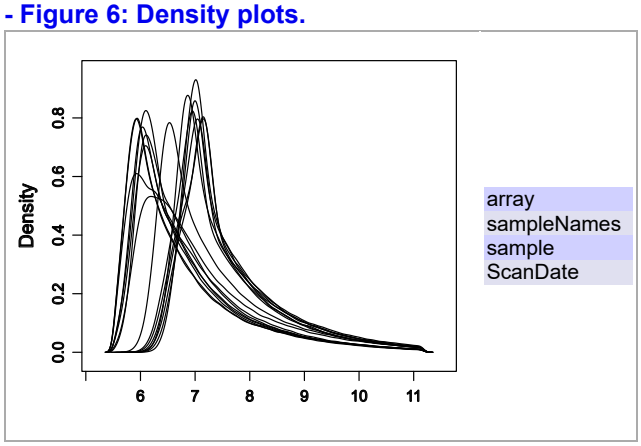


Figure 6 (PDF file) shows density estimates (smoothed histograms) of the data. Typically, the distributions of the arrays should have similar shapes and ranges. Arrays whose distributions are very different from the others should be considered for possible problems. Various features of the distributions can be indicative of quality related phenomena. For instance, high levels of background will shift an array's distribution to the right. Lack of signal diminishes its right right tail. A bulge at the upper end of the intensity range often indicates signal saturation.

Section 3: Variance mean dependence

- Figure 7: Standard deviation versus rank of the mean.

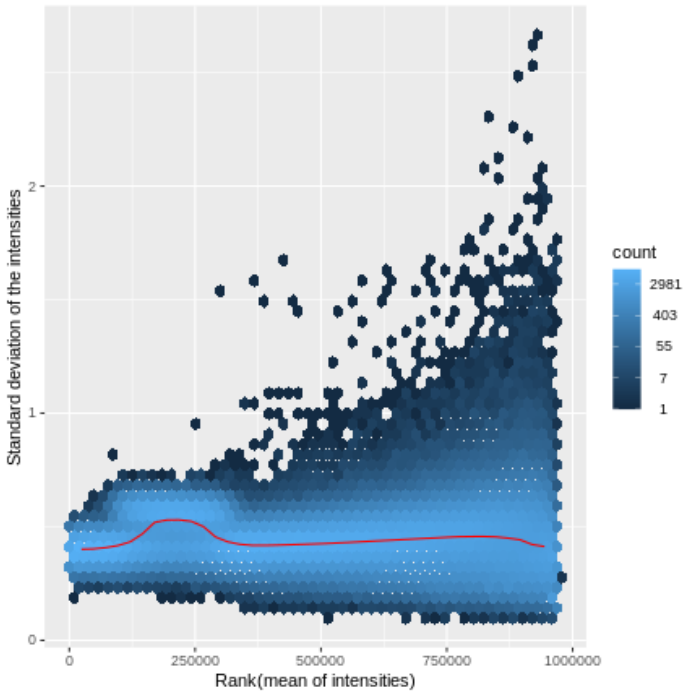


Figure 7 (PDF file) shows a density plot of the standard deviation of the intensities across arrays on the *y*-axis versus the rank of their mean on the *x*-axis. The red dots, connected by lines, show the running median of the standard deviation. After normalisation and transformation to a logarithm(-like) scale, one typically expects the red line to be approximately horizontal, that is, show no substantial trend. In some cases, a hump on the right hand of the x-axis can be observed and is symptomatic of a saturation of the intensities.

Section 4: Affymetrix specific plots

- Figure 8: Relative Log Expression (RLE).

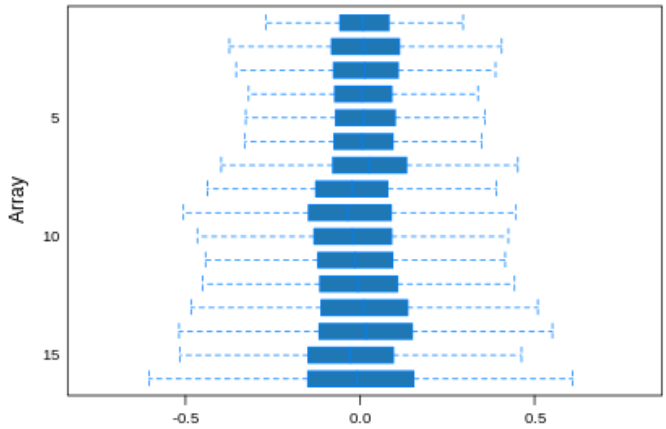


Figure 8 (PDF file) shows the *Relative Log Expression (RLE)* plot. Arrays whose boxes are centered away from 0 and/or are more spread out are potentially problematic. Outlier detection was performed by computing the Kolmogorov-Smirnov statistic R_a between each array's RLE values and the pooled, overall distribution of RLE values.

- Figure 9: Outlier detection for Relative Log Expression (RLE).

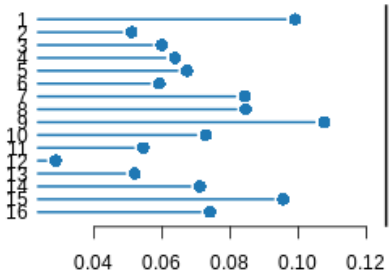


Figure 9 (PDF file) shows a bar chart of the Kolmogorov-Smirnov statistic R_a of the RLE values, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 0.126 was determined, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

- Figure 10: Normalized Unscaled Standard Error (NUSE).

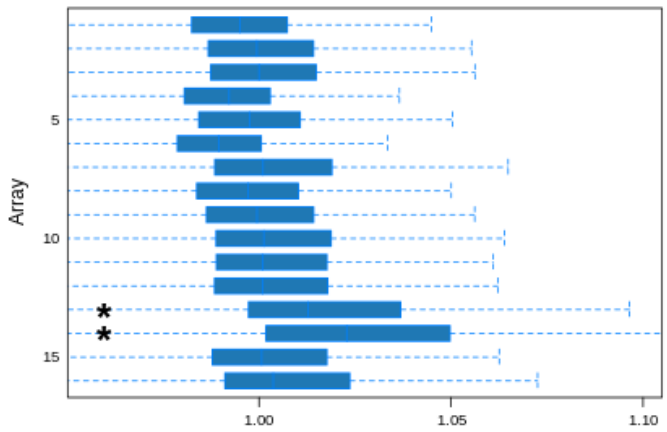


Figure 10 (PDF file) shows the *Normalized Unscaled Standard Error (NUSE)* plot. For each array, the boxes should be centered around 1. An array were the values are elevated relative to the other arrays is typically of lower quality. Outlier detection was performed by computing the 75% quantile N_a of each array's NUSE values and looking for arrays with large N_a.

- Figure 11: Outlier detection for Normalized Unscaled Standard Error (NUSE).

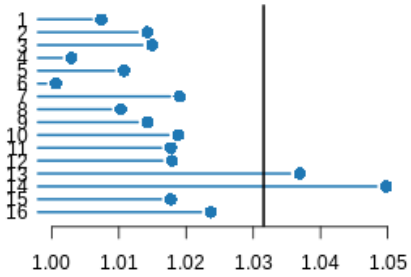


Figure 11 (PDF file) shows a bar chart of the N_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 1.03 was determined, which is indicated by the vertical line. 2 arrays exceeded the threshold and were considered outliers.

- Figure 12: RNA digestion plot.

array
sampleNames
sample
ScanDate

Figure 12 (PDF file) shows the *RNA digestion* plot. The shown values are computed from the preprocessed data (after background correction and quantile normalisation). Each array is represented by a single line; move the mouse over the lines to see their corresponding sample names. The plot can be used to identify array(s) that have a slope very different from the others. This could indicate that the RNA used for that array has been handled differently from what was done for the other arrays.

- Figure 13: Perfect matches and mismatches.

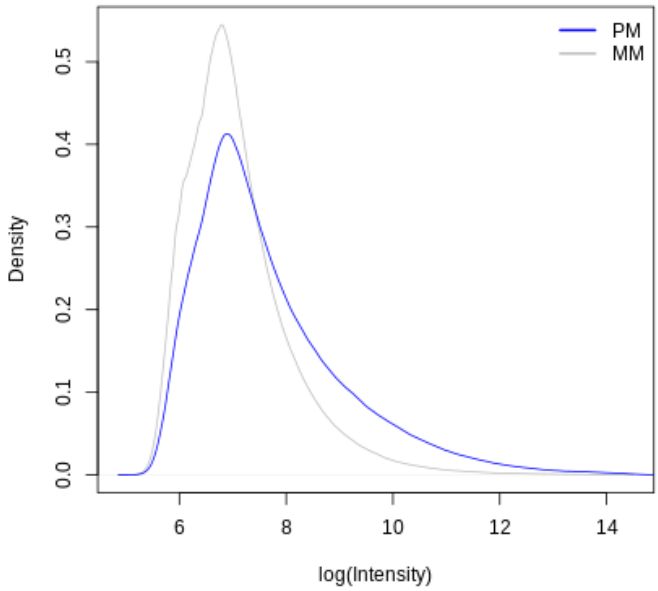


Figure shows the density distributions of the $\log_2$ intensities grouped by the matching type of the probes. The blue line shows a density estimate (smoothed histogram) from intensities of perfect match probes (PM), the grey line, one from the mismatch probes (MM). We expect that MM probes have poorer hybridization than PM probes, and thus that the PM curve be to the right of the MM curve.

Section 5: Individual array quality

- Figure 14: MA plots.

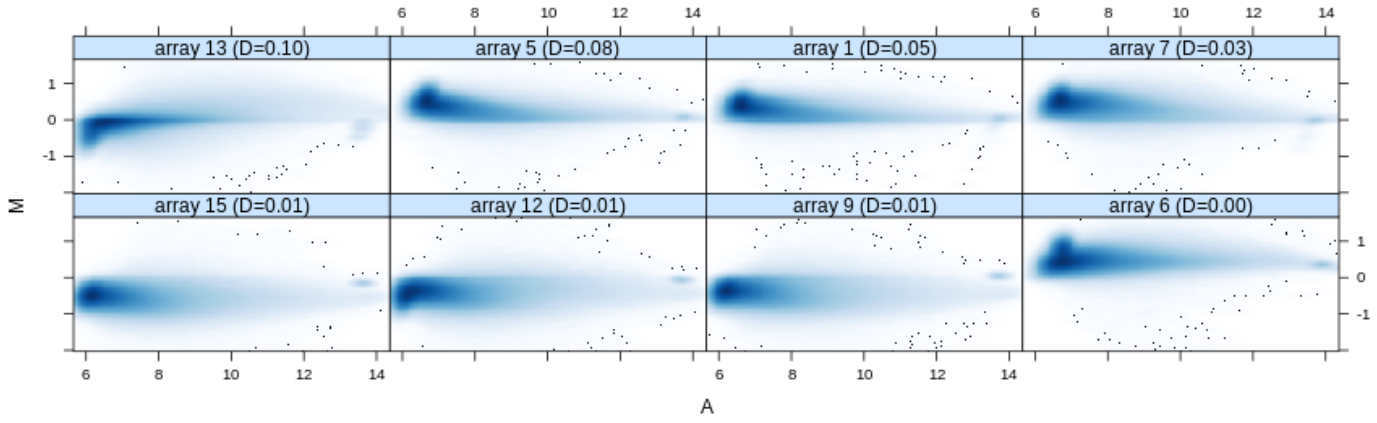


Figure 14 (PDF file) shows MA plots. M and A are defined as:
$M = \log_2(I_1) - \log_2(I_2)$
$A = 1/2\ (\log_2(I_1)+\log_2(I_2))$,
where I_1 is the intensity of the array studied,and I_2 is the intensity of a "pseudo"-array that consists of the median across arrays. Typically, we expect the mass of the distribution in an MA plot to be concentrated along the M = 0 axis, and there should be no trend in M as a function of A. If there is a trend in the lower range of A, this often indicates that the arrays have different background intensities; this may be addressed by background correction. A trend in the upper range of A can indicate saturation of the measurements; in mild cases, this may be addressed by non-linear normalisation (e.g. quantile normalisation).
Outlier detection was performed by computing Hoeffding's statistic D_a on the joint distribution of A and M for each array. Shown are first the 4 arrays with the highest values of D_a, then the 4 arrays with the lowest values. The value of D_a is shown in the panel headings. 0 arrays had D_a>0.15 and were marked as outliers. For more information on Hoeffing's D-statistic, please see the manual page of the function `hoeffd` in the `Hmisc` package.

- Figure 15: Outlier detection for MA plots.

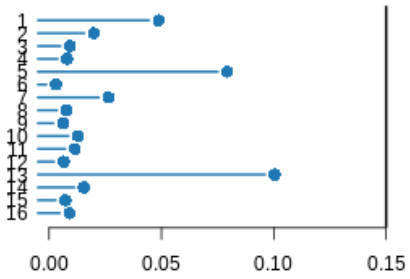


Figure 15 (PDF file) shows a bar chart of the D_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. A threshold of 0.15 was used, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

- Figure 16: Spatial distribution of M.

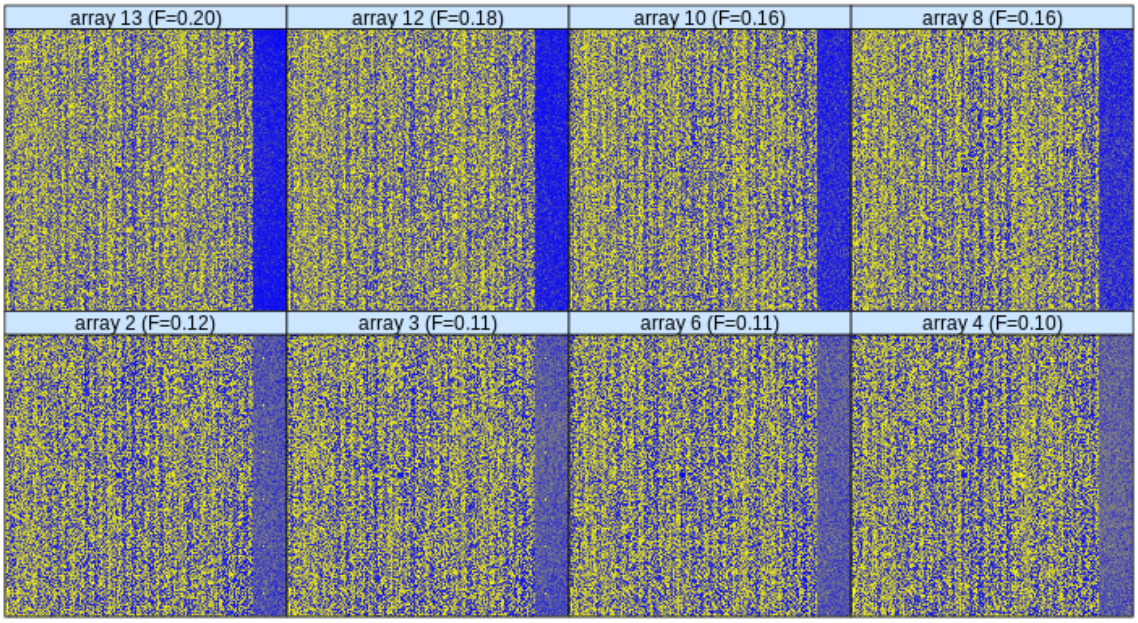


Figure 16 (PDF file) shows false color representations of the arrays' spatial distributions of feature intensities (M). Normally, when the features are distributed randomly on the arrays, one expects to see a uniform distribution; control features with particularly high or low intensities may stand out. The color scale is proportional to the ranks of the probe intensities . Note that the rank scale has the potential to amplify patterns that are small in amplitudebut systematic within an array. It is possible to switch off the rank scaling by modifying the argument`scale` in the call of the `aqm.spatial` function.
Outlier detection was performed by computing F_a , the sum of the absolutes value of low frequency Fourier coefficients, as a measure of large scale spatial structures. Shown are first the 4 arrays with the highest values of F_a, then the 4 arrays with the lowest values. The value of F_a is shown in the panel headings.

- Figure 17: Outlier detection for Spatial distribution of M.

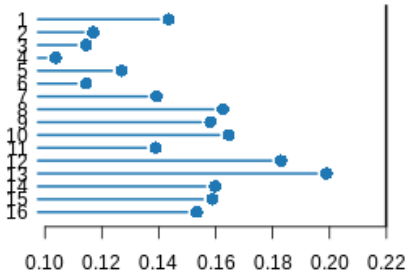


Figure 17 (PDF file) shows a bar chart of the F_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 0.22 was determined, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

This report has been created with arrayQualityMetrics 3.42.0 under R version 3.6.2 (2019-12-12).

(Page generated on Thu Apr 9 00:03:30 2020 by hwriter)

Annex 2.2 - Quality Control of the PSM microarray datasets

arrayQualityMetrics report for affybatch_psm

- Array metadata and outlier detection overview

	array	sampleNames	*1	*2	*3	*4	*5	*6	sample	ScanDate
☐	1	Gga01.CEL							1	06/02/05 11:57:57
☐	2	Gga02.CEL							2	07/01/05 11:46:03
☐	3	Gga03.CEL							3	07/01/05 11:56:46
☐	4	Gga04.CEL							4	07/07/05 11:39:28
☐	5	Gga05.CEL							5	07/14/05 11:17:00
☐	6	Gga06.CEL							6	07/12/05 11:18:20
☐	7	Gga07.CEL							7	07/14/05 11:06:25
☐	8	Gga08.CEL							8	07/07/05 11:28:21
☐	9	Gga09.CEL							9	07/07/05 11:07:30
☐	10	Gga10.CEL							10	07/12/05 11:28:46
☐	11	Gga11.CEL							11	07/12/05 11:08:10
☐	12	Gga12.CEL							12	07/01/05 11:35:36
☐	13	Gga13.CEL							13	07/14/05 10:55:56
☐	14	Gga14.CEL							14	06/02/05 12:09:01
☐	15	Gga15.CEL							15	07/01/05 12:07:04
☐	16	Gga16.CEL							16	07/07/05 11:17:59
☐	17	Gga17.CEL							17	07/19/05 11:18:50
☐	18	Gga18.CEL							18	07/19/05 11:29:13
☐	19	GSM1968008_OPE_PSM1dup1_Chicken.CEL.gz							19	12/10/10 09:57:47
☐	20	GSM1968009_OPE_PSM1dup2_Chicken.CEL.gz							20	12/15/10 11:10:48
☑	21	GSM1968010_OPE_PSM2dup1_Chicken.CEL.gz					x		21	12/10/10 10:52:23
☐	22	GSM1968011_OPE_PSM2dup2_Chicken.CEL.gz							22	12/15/10 10:53:01
☐	23	GSM1968012_OPE_PSM3dup1_Chicken.CEL.gz							23	12/10/10 10:34:25
☐	24	GSM1968013_OPE_PSM3dup2_Chicken.CEL.gz							24	12/15/10 10:07:50
☑	25	GSM1968014_OPE_PSM4dup1_Chicken.CEL.gz					x		25	12/10/10 10:25:12
☑	26	GSM1968015_OPE_PSM4dup2_Chicken.CEL.gz					x		26	12/15/10 10:25:47
☑	27	GSM1968016_OPE_PSM5dup1_Chicken.CEL.gz					x		27	12/10/10 10:16:03
☐	28	GSM1968017_OPE_PSM6dup1_Chicken.CEL.gz							28	12/10/10 10:43:35
☑	29	GSM1968018_OPE_PSM6dup2_Chicken.CEL.gz					x		29	12/15/10 12:28:54
☑	30	GSM1968019_OPE_PSM7dup1_Chicken.CEL.gz					x		30	12/10/10 09:48:34
☐	31	GSM1968020_OPE_PSM7dup2_Chicken.CEL.gz							31	12/15/10 12:37:47
☐	32	GSM1968021_OPE_PSM8dup1_Chicken.CEL.gz							32	12/15/10 11:01:48
☑	33	GSM1968022_OPE_PSM8dup2_Chicken.CEL.gz				x	x		33	12/15/10 12:46:34

The columns named *1, *2, ... indicate the calls from the different outlier detection methods:

1. outlier detection by Distances between arrays
2. outlier detection by Boxplots
3. outlier detection by Relative Log Expression (RLE)
4. outlier detection by Normalized Unscaled Standard Error (NUSE)
5. outlier detection by MA plots

6. outlier detection by Spatial distribution of M

The outlier detection criteria are explained below in the respective sections. Arrays that were called outliers by at least one criterion are marked by checkbox selection in this table, and are indicated by highlighted lines or points in some of the plots below. By clicking the checkboxes in the table, or on the corresponding points/lines in the plots, you can modify the selection. To reset the selection, reload the HTML page in your browser.

At the scope covered by this software, outlier detection is a poorly defined question, and there is no 'right' or 'wrong' answer. These are hints which are intended to be followed up manually. If you want to automate outlier detection, you need to limit the scope to a particular platform and experimental design, and then choose and calibrate the metrics used.

Section 1: Between array comparison

- Figure 1: Distances between arrays.

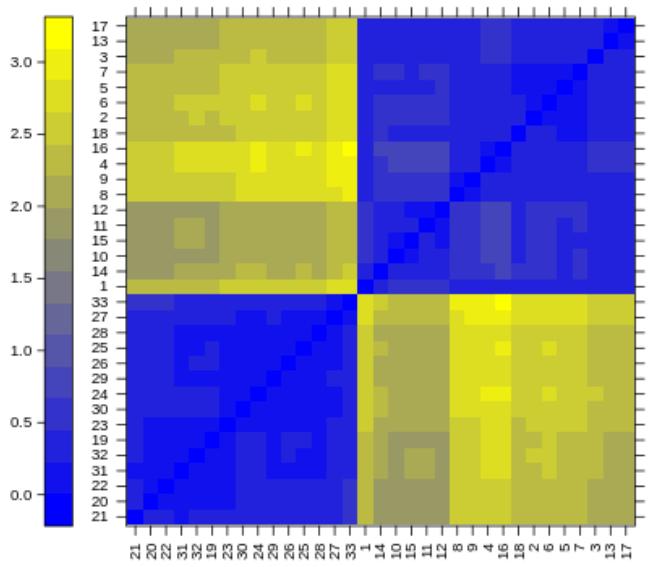


Figure 1 (PDF file) shows a false color heatmap of the distances between arrays. The color scale is chosen to cover the range of distances encountered in the dataset. Patterns in this plot can indicate clustering of the arrays either because of intended biological or unintended experimental factors (batch effects). The distance d_{ab} between two arrays a and b is computed as the mean absolute difference (L_1-distance) between the data of the arrays (using the data from all probes without filtering). In formula, d_{ab} = mean | M_{ai} - M_{bi} |, where M_{ai} is the value of the i-th probe on the a-th array. Outlier detection was performed by looking for arrays for which the sum of the distances to all other arrays, $S_a = \Sigma_b\, d_{ab}$ was exceptionally large. No such arrays were detected.

- Figure 2: Outlier detection for Distances between arrays.

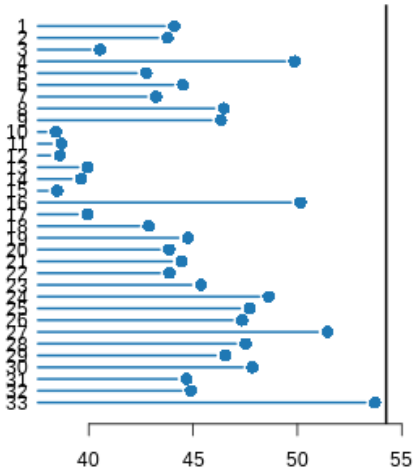


Figure 2 (PDF file) shows a bar chart of the sum of distances to other arrays S_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 54.3 was determined, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

- Figure 3: Principal Component Analysis.

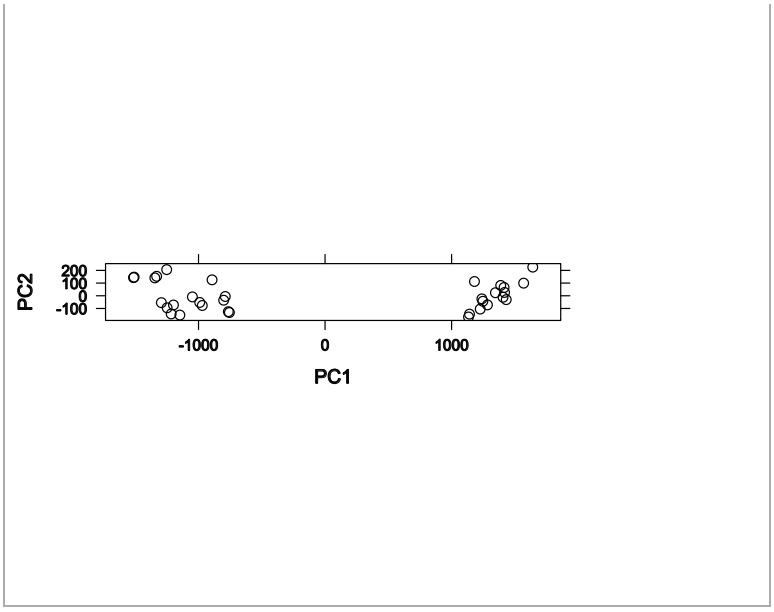


Figure 3 (PDF file) shows a scatterplot of the arrays along the first two principal components. You can use this plot to explore if the arrays cluster, and whether this is according to an intended experimental factor (you can indicate such a factor by color using the 'intgroup' argument), or according to unintended causes such as batch effects. Move the mouse over the points to see the sample names.
Principal component analysis is a dimension reduction and visualisation technique that is here used to project the multivariate data vector of each array into a two-dimensional plot, such that the spatial arrangement of the points in the plot reflects the overall data (dis)similarity between the arrays.

Section 2: Array intensity distributions

- Figure 4: Boxplots.

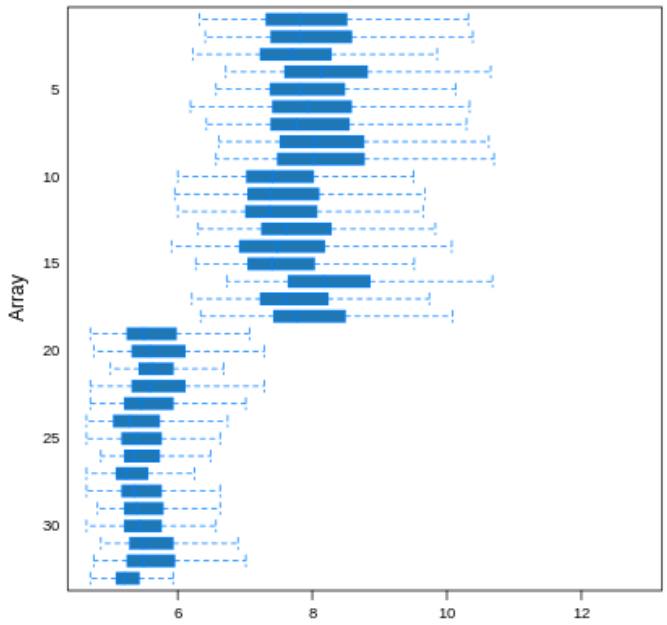


Figure 4 (PDF file) shows boxplots representing summaries of the signal intensity distributions of the arrays. Each box corresponds to one array. Typically, one expects the boxes to have similar positions and widths. If the distribution of an array is very different from the others, this may indicate an experimental problem. Outlier detection was performed by computing the Kolmogorov-Smirnov statistic K_a between each array's distribution and the distribution of the pooled data.

- Figure 5: Outlier detection for Boxplots.

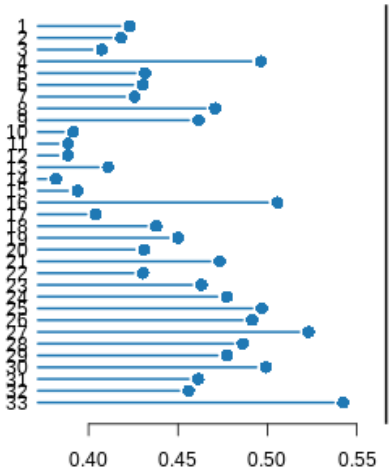


Figure 5 (PDF file) shows a bar chart of the Kolmogorov-Smirnov statistic K_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 0.566 was determined, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

- Figure 6: Density plots.

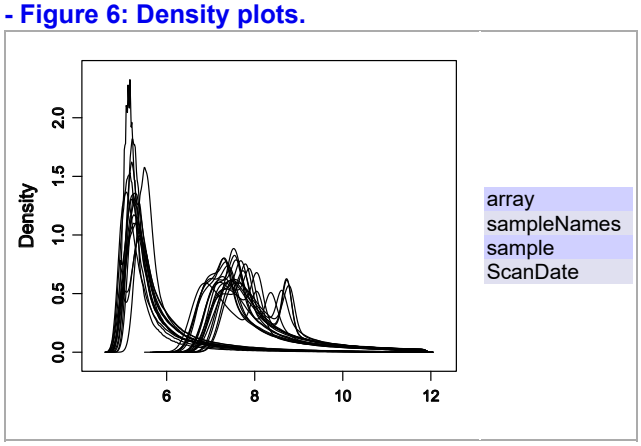


Figure 6 (PDF file) shows density estimates (smoothed histograms) of the data. Typically, the distributions of the arrays should have similar shapes and ranges. Arrays whose distributions are very different from the others should be considered for possible problems. Various features of the distributions can be indicative of quality related phenomena. For instance, high levels of background will shift an array's distribution to the right. Lack of signal diminishes its right right tail. A bulge at the upper end of the intensity range often indicates signal saturation.

Section 3: Variance mean dependence

- Figure 7: Standard deviation versus rank of the mean.

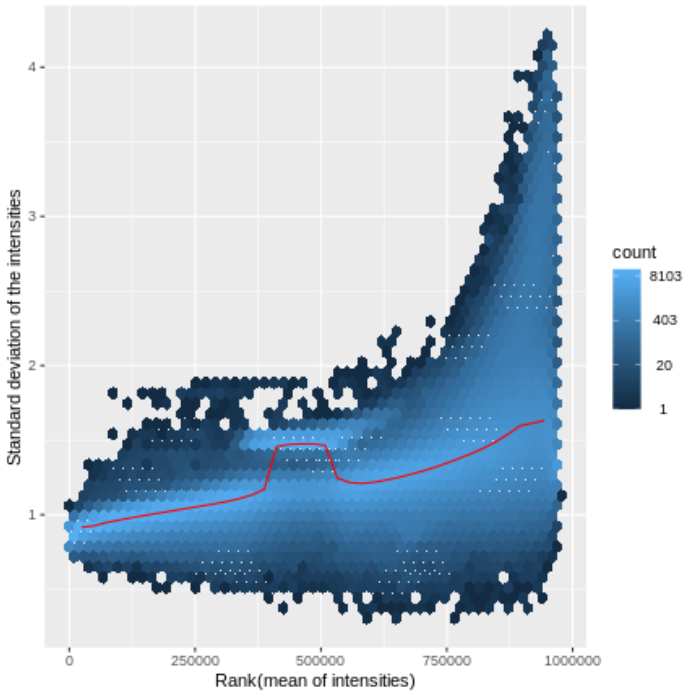


Figure 7 (PDF file) shows a density plot of the standard deviation of the intensities across arrays on the *y*-axis versus the rank of their mean on the *x*-axis. The red dots, connected by lines, show the running median of the standard deviation. After normalisation and transformation to a logarithm(-like) scale, one typically expects the red line to be approximately horizontal, that is, show no substantial trend. In some cases, a hump on the right hand of the x-axis can be observed and is symptomatic of a saturation of the intensities.

Section 4: Affymetrix specific plots

- Figure 8: Relative Log Expression (RLE).

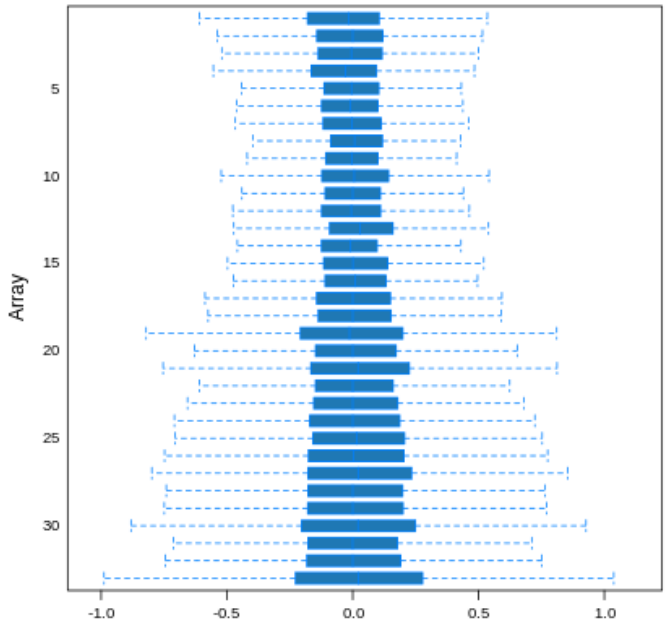


Figure 8 (PDF file) shows the *Relative Log Expression (RLE)* plot. Arrays whose boxes are centered away from 0 and/or are more spread out are potentially problematic. Outlier detection was performed by computing the Kolmogorov-Smirnov statistic R_a between each array's RLE values and the pooled, overall distribution of RLE values.

- Figure 9: Outlier detection for Relative Log Expression (RLE).

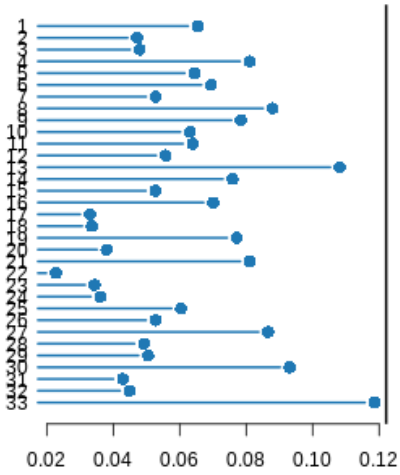


Figure 9 (PDF file) shows a bar chart of the Kolmogorov-Smirnov statistic R_a of the RLE values, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 0.122 was determined, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

- Figure 10: Normalized Unscaled Standard Error (NUSE).

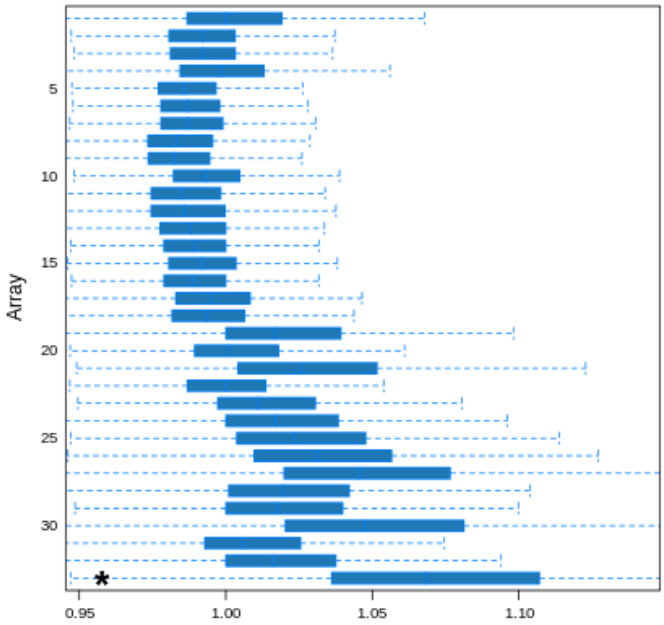


Figure 10 (PDF file) shows the *Normalized Unscaled Standard Error (NUSE)* plot. For each array, the boxes should be centered around 1. An array were the values are elevated relative to the other arrays is typically of lower quality. Outlier detection was performed by computing the 75% quantile N_a of each array's NUSE values and looking for arrays with large N_a.

- Figure 11: Outlier detection for Normalized Unscaled Standard Error (NUSE).

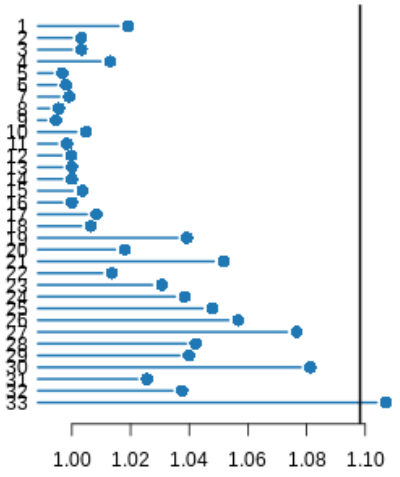


Figure 11 (PDF file) shows a bar chart of the N_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 1.1 was determined, which is indicated by the vertical line. One array exceeded the threshold and was considered an outlier.

- Figure 12: RNA digestion plot.

array
sampleNames
sample
ScanDate

Figure 12 (PDF file) shows the *RNA digestion* plot. The shown values are computed from the preprocessed data (after background correction and quantile normalisation). Each array is represented by a single line; move the mouse over the lines to see their corresponding sample names. The plot can be used to identify array(s) that have a slope very different from the others. This could indicate that the RNA used for that array has been handled differently from what was done for the other arrays.

- Figure 13: Perfect matches and mismatches.

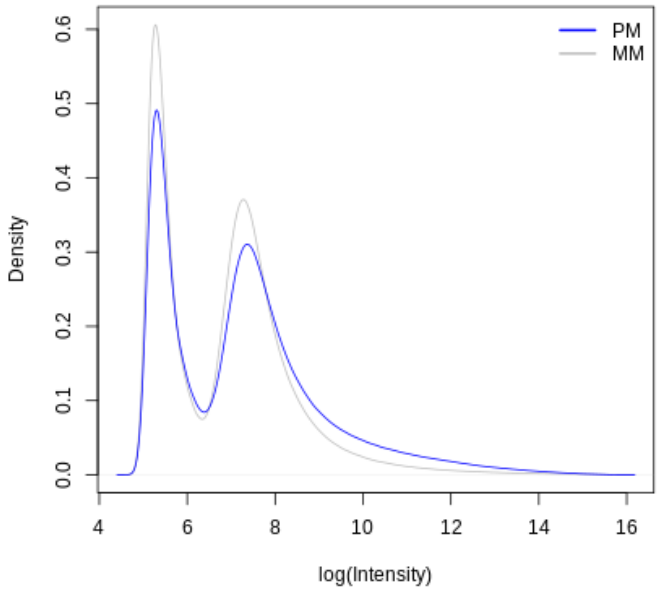


Figure shows the density distributions of the $\log_2$ intensities grouped by the matching type of the probes. The blue line shows a density estimate (smoothed histogram) from intensities of perfect match probes (PM), the grey line, one from the mismatch probes (MM). We expect that MM probes have poorer hybridization than PM probes, and thus that the PM curve be to the right of the MM curve.

Section 5: Individual array quality

- Figure 14: MA plots.

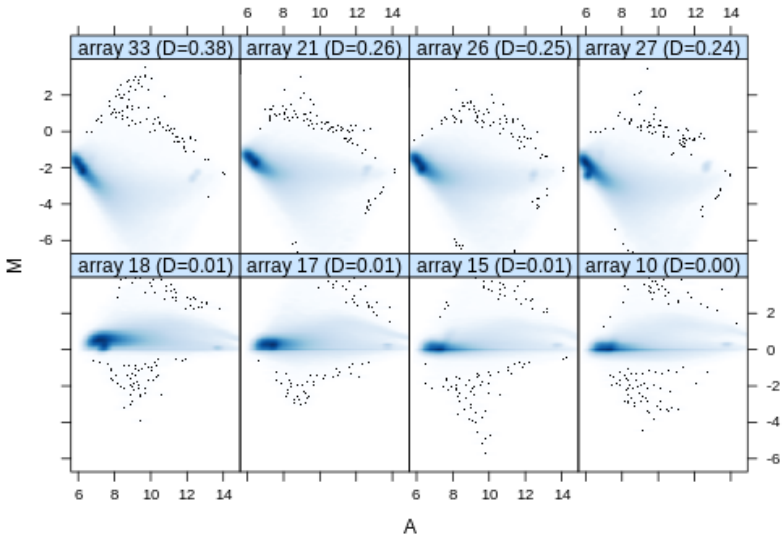


Figure 14 (PDF file) shows MA plots. M and A are defined as:
$M = \log_2(I_1) - \log_2(I_2)$
$A = 1/2\ (\log_2(I_1)+\log_2(I_2))$,
where I_1 is the intensity of the array studied,and I_2 is the intensity of a "pseudo"-array that consists of the median across arrays. Typically, we expect the mass of the distribution in an MA plot to be concentrated along the M = 0 axis, and there should be no trend in M as a function of A. If there is a trend in the lower range of A, this often indicates that the arrays have different background intensities; this may be addressed

by background correction. A trend in the upper range of A can indicate saturation of the measurements; in mild cases, this may be addressed by non-linear normalisation (e.g. quantile normalisation).
Outlier detection was performed by computing Hoeffding's statistic D_a on the joint distribution of A and M for each array. Shown are first the 4 arrays with the highest values of D_a, then the 4 arrays with the lowest values. The value of D_a is shown in the panel headings. 7 arrays had D_a>0.15 and were marked as outliers. For more information on Hoeffing's D-statistic, please see the manual page of the function `hoeffd` in the `Hmisc` package.

- Figure 15: Outlier detection for MA plots.

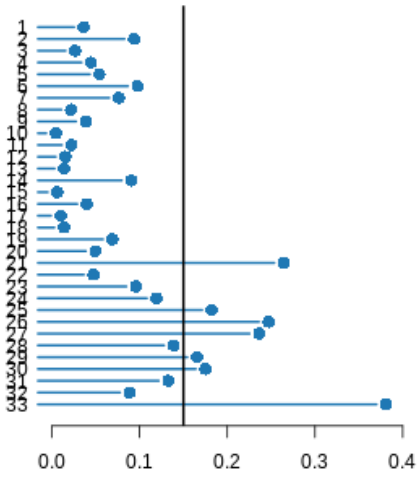


Figure 15 (PDF file) shows a bar chart of the D_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. A threshold of 0.15 was used, which is indicated by the vertical line. 7 arrays exceeded the threshold and were considered outliers.

- Figure 16: Spatial distribution of M.

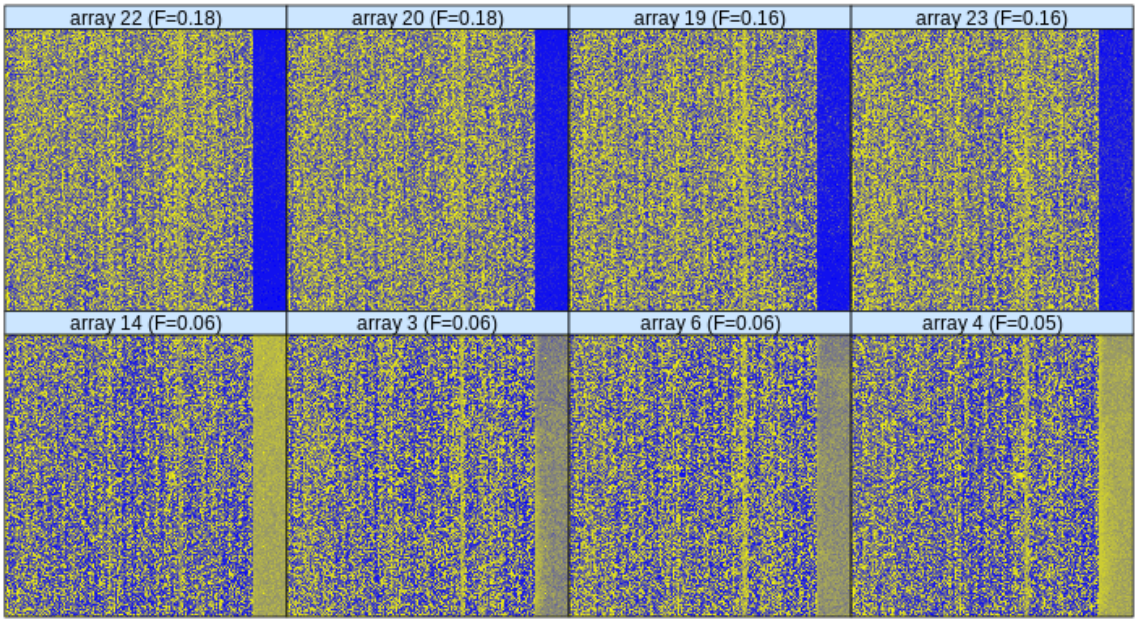


Figure 16 (PDF file) shows false color representations of the arrays' spatial distributions of feature intensities (M). Normally, when the features are distributed randomly on the arrays, one expects to see a uniform distribution; control features with particularly high or low intensities may stand out. The color scale is proportional to the ranks of the probe intensities . Note that the rank scale has the potential to amplify patterns that are small in amplitudebut systematic within an array. It is possible to switch off the rank scaling by modifying the argument`scale` in the call of the `aqm.spatial` function.
Outlier detection was performed by computing F_a , the sum of the absolutes value of low frequency Fourier coefficients, as a measure of large scale spatial structures. Shown are first the 4 arrays with the highest values of F_a, then the 4 arrays with the lowest values. The value of F_a is shown in the panel headings.

- Figure 17: Outlier detection for Spatial distribution of M.

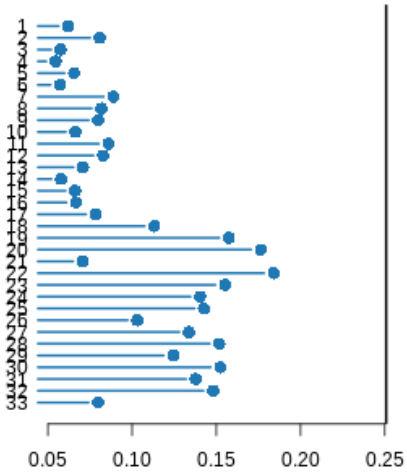


Figure 17 (PDF file) shows a bar chart of the F_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 0.251 was determined, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

This report has been created with arrayQualityMetrics 3.42.0 under R version 3.6.2 (2019-12-12).

(Page generated on Mon Apr 13 17:01:16 2020 by hwriter)

Annex 3 | Publications of the “FrozenChicken” RData package

Annex 3.1 - Publication of the “FrozenChicken” RData package in the bioRxiv journal

FrozenChicken: Promoting the meta-analysis of chicken microarray data

by Isabel Duarte, Marta Liber*, Ramiro Magno, and Raquel P. Andrade*

Abstract The FrozenChicken RData package, contains the frozen vectors for the commercially available (in situ oligonucleotide) Affymetrix Chicken Genome Array (GEO platform id GPL3213). This package will promote, simplify, and ease the meta-analysis of chicken microarray data by the research community studying vertebrate development using the chick model organism. The package is freely available in https://github.com/iduarte/FrozenChicken. (**Equal contribution*.)

Introduction

Background | *Gallus gallus* (chicken) is one of the most valuable model organisms for the study of the vertebrate embryo development. Such studies can be aided by pooling together OMICs data from public repositories, like GEO (Gene Expression Omnibus) and ArrayExpress, that currently contain more than 11.660 datasets from chicken, representing a wealth of data that can be explored to answer fundamental questions and generate new hypotheses. However, since these data come from different experiments, their meta-analysis requires proper normalization to deal with the technical biases and batch effects before making the data comparable for statistical analysis.

Approach | An effective method for such normalization is the single-array pre-processing provided by the Frozen Robust Multiarray Analysis (**fRMA**) (McCall MN, et al. Biostatistics. 2010). This method uses *"frozen" RMA* vectors pre-computed from great amounts of available data for the same microarray platform, accounting for the aforementioned biases. However, such frozen vectors/parameters are available for multiple organisms (most notably, human, mouse, zebrafish, and fruit-fly among others), but not for chicken, hence preventing, or delaying, the proper meta-analysis of chicken transcriptomics datasets without prior computation of self-generated frozen parameters.

Output | Here, we present the RData package **FrozenChicken** containing the chicken microarray frozen vectors that can be directly plugged-in to a chicken microarray analysis pipeline (using fRMA) without any other prior data gathering and processing. The package is freely available for the research community at: https://github.com/iduarte/FrozenChicken. A version of this article in the form of an html tutorial has been deposited in zenodo DOI:10.5281/zenodo.3765944.

Significance | This package will directly benefit the chicken research community by facilitating future meta-analysis studies using transcriptomics datasets from public repositories, hence directly contributing to the quality of the scientific research using the chick model organism.

Methods

Methods | 1. Data Collection

1 | Search for relevant GEO data series
The chicken microarray datasets used were gathered using the **ESearch** function from the **Entrez Programming Utilities** (E-utilities) that provide a programmatic connection with the Entrez query system from NCBI.

This search returned a list of Unique Identifiers (UIDs) for **1739 records** that met the following query criteria:
- search the database Geo DataSets (*gds*);
- search for GEO platform id GPL3213, which is the **Affymetrix Chicken Genome Array** (the chicken commercially available microarray chip);
- select only records that have *.CEL* supplementary files available for downloading.

The issued query was the following:
https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=gds&term=GPL3213%5BACCN%5D+AND+cel%5BsuppFile%5D&retmax=5000&usehistory=y

2 | Gather summary data for the data series found
Using the list of UIDs returned from the previous step, we gathered the metadata associated with each entry. For this we used the **ESummary** function from NCBI's **E-utilities**, that returns the documented summaries for each UID, including the Geo Series (GSE) identifier for each experiment. The ESummary tool directly uses the esearch URL retrieved from the previous step.

3 | Download the relevant data sets
Careful manual inspection of the results retrieved from the previous step, led to the collection of

128 relevant GSE records. These were downloaded using the R package `GEOquery` (version 2.50.5) (https://academic.oup.com/bioinformatics/article/23/14/1846/190290).

```
## Load the required package GEOquery
library (GEOquery)

## Set the working directory to the folder where the downlods will be saved
setwd("/home/FrozenChicken/data")

## Create the vector of GSE ids to download
GSE.ids <- c("GSE94622","GSE114476","GSE89325","GSE106788","GSE106787",
             "GSE109451","GSE81023","GSE87663","GSE79963","GSE69862",
             "GSE71888","GSE87486","GSE51330","GSE76794","GSE82344",
             "GSE48454","GSE48453","GSE69684","GSE81994","GSE39450",
             "GSE81717","GSE81461","GSE75798","GSE35430","GSE14220",
             "GSE71117","GSE62882","GSE53932","GSE53931","GSE53930",
             "GSE33389","GSE60754","GSE31507","GSE59002","GSE59921",
             "GSE59920","GSE48359","GSE48116","GSE52227","GSE34687",
             "GSE14587","GSE44394","GSE50880","GSE22222","GSE47191",
             "GSE38168","GSE31508","GSE31524","GSE31506","GSE31505",
             "GSE31501","GSE31499","GSE31476","GSE37070","GSE42845",
             "GSE39602","GSE42516","GSE40802","GSE40100","GSE398242",
             "GSE39346","GSE27958","GSE35581","GSE38381","GSE38107",
             "GSE37782","GSE35413","GSE15830","GSE32272","GSE21706",
             "GSE25185","GSE25151","GSE32494","GSE24641","GSE25588",
             "GSE29565","GSE29564","GSE29563","GSE29562","GSE28634",
             "GSE28391","GSE28388","GSE21915","GSE22592","GSE23592",
             "GSE23881","GSE23389","GSE19698","GSE22230","GSE17758",
             "GSE17725","GSE21679","GSE15143","GSE15141","GSE14489",
             "GSE14013","GSE14509","GSE11636","GSE18477","GSE18778",
             "GSE18568","GSE18506","GSE126752","GSE16081","GSE16064",
             "GSE15413","GSE15382","GSE9251","GSE11597","GSE10538",
             "GSE12268","GSE11439","GSE8010","GSE8483","GSE8018",
             "GSE8017","GSE8016","GSE10231","GSE8495","GSE9884",
             "GSE8693","GSE7805","GSE6543","GSE7176","GSE6856",
             "GSE6844","GSE6843","GSE6868")

## Run getGEOSuppFiles function from the GEOquery function
sapply(GSE.ids, getGEOSuppFiles)
```

Methods | 2. Building the FrozenChicken Package

To create the FrozenChicken R package we used the `frmaTools` R package (version 1.34.0) (https://doi.org/10.1186/1471-2105-12-369). This package requires the usage of the same number of samples per experiment in order not to create biases. We chose to use **4 samples** from each experiment, keeping only the datasets that had at least four microarrays.

Therefore, we used **118 Geo DataSets** from Affymetrix Chicken Genome Array platform, from which 4 arrays, per batch, were randomly selected (**Batch number = 118** and **Batch size = 4**). This approach retrieved a powerful set of **472 chips** that were used as input for the `frmaTools` R package to compute the frozen vectors/parameters.

```
## Load the required packages
library (frmaTools)
library (chickencdf)  # Annotation package for the chicken microarray

## Set the directories required
frma_data_dir <- "/home/FrozenChicken/data"
frma_output_dir <- "/home/FrozenChicken/output"

## Create a numerical vector for the batch size
frma_batch_size <- 4

## Create the batch-id string
fRMA.chiken.batch.id <- rep(1:(length(dir(frma_data_dir))/frma_batch_size),
```

```
                                each=frma_batch_size)

## Run the makeVectorPackage function from the frmaTools package
makeVectorPackage (dir(frma_data_dir), fRMA.chiken.batch.id,
                   file.dir=frma_data_dir, output.dir=frma_output_dir,
                   version="1.0", maintainer="Marta Liber <mliber.pt@gmail.com>",
                   species="Gallus gallus", annotation="chickencdf",
                   packageName="affyChickGenomeArrayfrmavecs", background="rma",
                   normalize="quantile", normVec=NULL, type="AffyBatch",
                   unlink=TRUE, verbose=TRUE)
```

Results

Case Study Using FrozenChicken

In this section we will show how to use the FrozenChicken vectors in a microarray data analysis of chicken transcriptomics. The workflow described here was performed in R, using RStudio (version 1.1.463). The typical workflow of a microarray data analysis is shown in Figure 1.

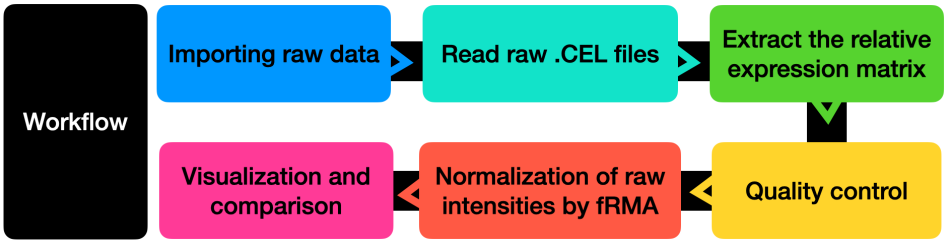


Figure 1: Microarray data analysis typical workflow.

Here, we will be looking at three chicken microarray datasets from different experiments:
- chick PSM tissue, embryo stage HH12, from GEO record GSE75798 (Oginuma, M. *et al.*, 2017);
- chick right PSM region, embryo stages HH11-12, from Array Express record E-MTAB-406 (Krol *et al.*, 2011);
- chick anterior and posterior limb bud at stage HH20, from ArrayExpress record E-MTAB-4048 (Anderson, C. *et al.*, 2016).

Case study | 1. Install FrozenChicken and Additional R Packages We will start by installing the FrozenChicken package, which is deposited in GitHub. To install it directly from GitHub you should use the package `remotes`. If you do not have it, install it first:

```
## Install the package from CRAN repository
install.packages("remotes")

## Load the package
library(remotes)
```

Then install the R package `FrozenChicken` directly from GitHub:

```
remotes::install_github("iduarte/FrozenChicken")
```

Next you can load the library named **affyChickGenomeArrayfrmavecs** and the *frozen parameters* become available for the normalization of chicken microarray data from different experiments (provided that all use the same Affymetrix Chicken Genome Array platform).

```
## Load the FrozenChicken package
  # This is the full name of the FrozenChicken data object
library(affyChickGenomeArrayfrmavecs)

## Load the affyChickGenomeArrayfrmavecs data set
data(affyChickGenomeArrayfrmavecs)
```

To complete this case study, the following R Packages are required:

```
## Install Bioconductor (if not already installed)
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install(version = "3.10")

## Install the required Packages (if not already installed)
BiocManager::install(c("ArrayExpress",
                       "GEOquery",
                       "Biobase",
                       "affy",
                       "arrayQualityMetrics",
                       "ggplot2",
                       "frma",
                       "devtools"))
```

Case Study | 2. Obtaining the Gene Expression Matrix To conduct a transcriptomics data analysis, one must obtain a gene expression matrix, i.e. a data table that reports the expression level measured for each gene. When the data to be analysed originates from microarrays that are deposited in public repositories, namely GEO or ArrayExpress, the completion of the following steps will generate a gene expression matrix:

1. Download the .CEL files (raw microarray data from Affymetrix) from its data repository. The data can be download using the `ArrayExpress` and `GEOquery` packages, respectively.
2. Read the raw .CEL files into R using the `affy` package, creating an 'AffyBatch' object containing the microarray data.
3. Extract the gene expression matrix from the 'AffyBatch' object.

```
## Load the required packages for this code chunk
library(devtools)
library(ArrayExpress)
library(GEOquery)
library(Biobase)
library(affy)

## Set up the directories used for the analysis
  ## (NOTE: Change the paths to the correct directories from your computer)
setwd ("/home/microarray_meta_analysis/")
data_dir <- "/home/microarray_meta_analysis/data"
output_dir <- "/home/microarray_meta_analysis/output"

## Step 1 - Downloading the .CEL files
setwd (data_dir)
getGEOSuppFiles ("GSE75798", makeDirectory = TRUE)
getAE ("E-MTAB-4048", type = "raw")
getAE ("E-MTAB-406", type = "raw")

## Step 2 - Load the .CEL files, i.e. import the raw data into R
affybatch_chick <- ReadAffy (celfile.path = data_dir)

## Step 3 - Extract the raw expression values using the exprs() function
expres_chick_raw <- exprs (affybatch_chick)

# Log2 transform (only if values are not log already)
  # NOTE: Code adapted from NCBI's GEO2R scripts.
qx <- as.numeric(quantile(expres_chick_raw,
                          c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
  (qx[6]-qx[1] > 50 && qx[2] > 0) ||
  (qx[2] > 0 && qx[2] < 1 && qx[4] > 1 && qx[4] < 2)
# log2 transform values if they are not in log scale already
if (LogC) {
  expres_chick_raw[which(expres_chick_raw <= 0)] <- NaN   # remove zeros
  expres_chick_raw <- log2(expres_chick_raw)
  cat("The RAW expression values were not log2 transformed,",
```

```
        "and now they have been log2 transformed.")
}

# Cleanup (delete unnecessary variables)
rm (qx, LogC)
```

Case Study | 3. Quality Control Quality control (QC) is an important step to remove data from faulty arrays. The `arrayQualityMetrics` package flags potential outliers and outputs plots to aid with the visual inspection of the results.

All arrays passed the QC criteria, and so all will be included in the next analysis steps. If any outlier were to be flagged, then those arrays should be removed from the analysis, and the quality control steps have to be re-run.

```
## Load the required library for quality control
library(arrayQualityMetrics)

## Step 3. Quality Control Report
  # This package uses the raw affybatch object directly
  # and not the expression matrix
  # (which is why we log transform the data).
arrayQualityMetrics(expressionset = affybatch_chick,
                    outdir = output_dir,
                    force = FALSE, do.logtransform = TRUE)
```

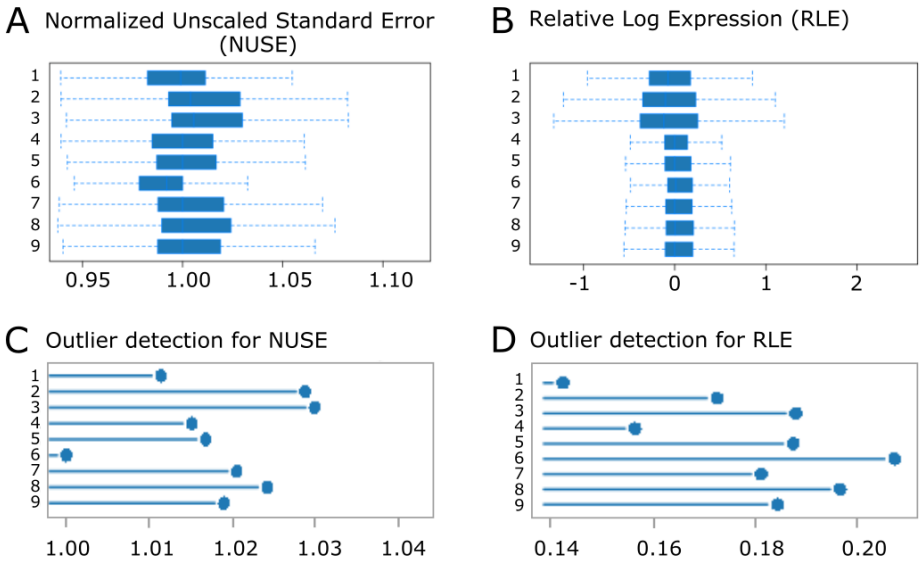


Figure 2: Quality Control metrics report. (A) Normalized Unscaled Standard Error (NUSE) plot. (B) Relative Log Expression (RLE) plot. (C) Bar chart representation for the outlier detection based on NUSE metrics in (A). (D) Bar chart representation for the outlier detection based on RLE metrics in (B).

Case Study | 4. Normalization with fRMA Using FrozenChicken The normalization of data obtained from different experiments is pivotal to make the data comparable between arrays. Using the frozenRMA method this can be easily done using a vector of frozen parameters pre-computed from diverse datasets from the same microarray chip. `FrozenChicken` presents a package containing the pre-computation of these frozen parameters for the chicken commercial microarray **Affymetrix Chicken Genome Array** to be used with the `fRMA` package.

```
## Load the required packages
library(frma)
library(affyChickGenomeArrayfrmavecs)

## Load the affyChickGenomeArrayfrmavecs data set to use latter
data(affyChickGenomeArrayfrmavecs)
```

```
## Step 4 - frozenRMA normalization using the FrozenChicken vectors
eset_chick_frma <- frma(affybatch_chick,
                        background="rma",
                        normalize="quantile",
                        summarize="robust_weighted_average",
                        target="probeset",
                        input.vecs=affyChickGenomeArrayfrmavecs,
                        output.param=NULL, verbose=FALSE)

# The data is Log2 tranformed by the process of fRMA normalization
expres_chick_frma <- exprs(eset_chick_frma)
```

Case Study | 5. Data Visualization Once the data have been normalized, we must confirm that the normalization was successful by running the quality control steps on the newly normalized data, and compare the results with the pre-normalized data. Here we show two of the most relevant plots to evaluate the success of the normalization procedure, namely, a boxplot (where each box corresponds to the intensity distribution of one array), and a Principal Component Analysis PCA plot to view the variation between the arrays (here, each dot is one array).

```
## Load the required package for this code chunk
library(ggplot2)

# Boxplots of raw log-intensity distribution
# requires the raw expression values extracted in step 2
boxplot(expres_chick_raw, col=c(rep("#b2df8a",3),
                                rep("#1f78b4",3),
                                rep("#fb9a99",3)),
        las = 3, cex.axis=0.75,
        main="Chicken Log2 raw expression values")

## PCA
pca_chicken <- prcomp(t(expres_chick_raw),
                      center = TRUE,
                      scale. = TRUE)
pca_chicken_information <- data.frame(pca_chicken$x,
                                      variance=as.numeric(round(
                                        100*summary(pca_chicken)$importance[2,],
                                        digits=2)),
                                   origin=c(rep("PSM AE", 3),
                                            rep("PSM GEO",3),
                                            rep("Limb AE",3)))
ggplot(pca_chicken_information,
       aes(x=pca_chicken_information[,1],
           y=pca_chicken_information[,2],
           color=origin)) +
  geom_point(size=2, alpha=0.7, show.legend = TRUE) + theme_bw() +
  labs(color='Origin') +
  scale_color_manual(values = c("#b2df8a", "#1f78b4", "#fb9a99")) +
  xlab(paste("PC1 (", pca_chicken_information$variance[1], "%)")) +
  ylab(paste("PC2 (", pca_chicken_information$variance[2], "%)")) +
  ggtitle("Chicken Log2 raw expression values") +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) -> pca_plot_chicken
pca_plot_chicken

## Boxplots of normalized log-intensity distribution
# requires the normalized expression values extracted in step 4
boxplot(expres_chick_frma,
        col=c(rep("#b2df8a",3),
              rep("#1f78b4",3),
              rep("#fb9a99",3)),
        las = 3, cex.axis=0.75,
        main="Chicken Log2 normalized expression values")
```

```
# PCA: provides another view of the correlations of expression between arrays.
pca_chicken_norm <- prcomp(t(expres_chick_frma),
                           center = TRUE,
                           scale. = TRUE)
pca_chicken_norm_information <- data.frame(pca_chicken_norm$x,
                                           variance=as.numeric(round(
                                             100*summary(pca_chicken_norm)$importance[2,],
                                             digits=2)),
                                           origin=c(rep("PSM AE", 3),
                                                    rep("PSM GEO",3),
                                                    rep("Limb AE",3)))

ggplot(pca_chicken_norm_information,
       aes(x=pca_chicken_norm_information[,1],
           y=pca_chicken_norm_information[,2],
           color=origin)) +
  geom_point(size=2, alpha=0.7, show.legend = TRUE) +
  theme_bw() +
  labs(color='Origin') +
  scale_color_manual(values=c("#b2df8a", "#1f78b4", "#fb9a99")) +
  xlab(paste("PC1 (", pca_chicken_norm_information$variance[1], "%)")) +
  ylab(paste("PC2 (", pca_chicken_norm_information$variance[2], "%)")) +
  ggtitle("Chicken Log2 normalized expression values") +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) -> pca_plot_chicken_norm
pca_plot_chicken_norm
```

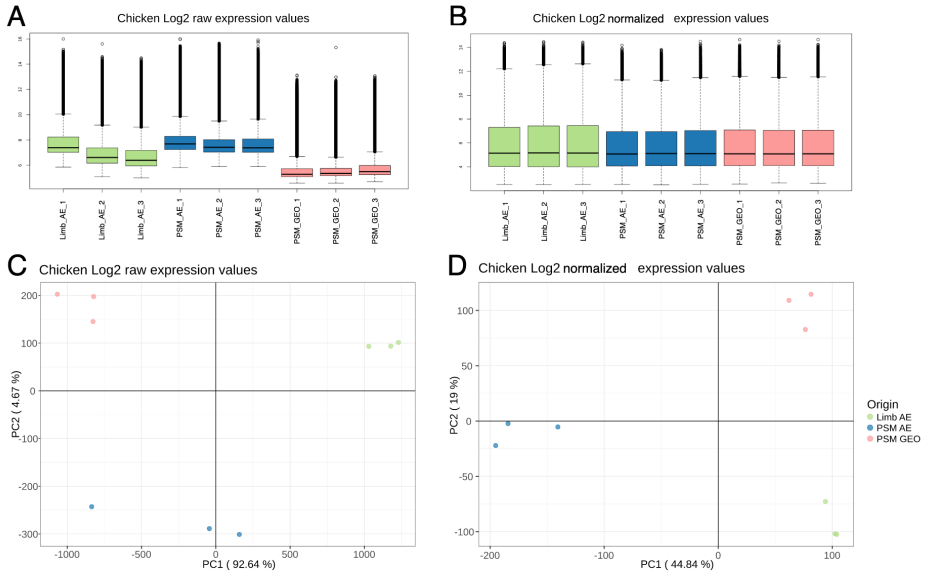


Figure 3: Comparison between raw and normalized gene expression values. (A) Boxplot of raw log2-transformed expression values. (B) Boxplot of normalized log2-transformed expression values. (C) PCA of raw log2-transformed expression values. (D) PCA of normalized log2-transformed expression values.

At this point, the data is log2-transformed and normalized, therefore, further analyses on the transcriptomics dataset may be performed (e.g. differential gene expression, and functional enrichment).

FrozenChicken Performance Evaluation

Before normalization, samples show variation between and within batches (Figure 3A). Additionally, PCA analysis found that 92.64% of variance between the data points is explained by the identity of the

experiment (Figure 3C). Thus, the major source of variation in the raw intensity measurements is due to batch effects that should be reduced after the fRMA normalization.

Since the purpose of normalization is to remove unwanted variation between the transcriptional profiles, we expect that after the normalization, the relative gene-expression estimates will be distributed in a homogeneous way across the arrays, and also, the variance found by the PCA will decrease.

Our results show that, after normalizing the samples with the frozen vectors from `FrozenChicken`, the arrays exhibit similar distribution profiles (Figure 3B), indicating that the normalization was successful.

In the PCA analysis, as expected, the variance described by the first component (PC1) has now decreased to 44.84% (Figure 3D). Additionally, the distances between the points in the first principal component from the raw data, range between -1500 and 1500 (Figure 3C), while the distances for the normalized values has decreased by nearly 10 fold (ranging between -200 and 100 (Figure 3D), further confirming the success of the fRMA normalization using the pre-computed parameters from `FrozenChicken`.

It should be noted that, despite the successful normalization, there are still variation in the data (Figure 3D), mostly explained by the difference in tissue types (Figure 3D and Figure 4B), i.e. the biological variability that we are interested in studying. In the PCA from the raw data, the samples cluster by data repository, showing that the major source of variation explained by the first component was the experiment (technical variation that we are not interested in studying) (Figure 3C and Figure 4A).

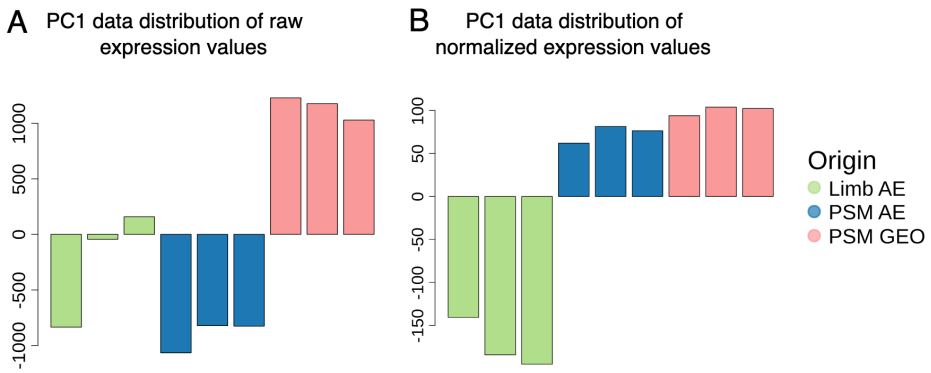


Figure 4: Distribution of distances calculated by PCA along its first component. (A) Distribution of distances along the first component of PCA on raw expression values. (B) Distribution of distances along the first component of PCA on normalized expression values.

Conclusion

This case study shows that `FrozenChicken` is a **reliable data package** to be used with `fRMA` normalization for the pre-processing steps of chicken microarray data from different experiments, therefore promoting, simplifying, and easing future meta-analyses of chicken transcriptomics datasets from public repositories. This package will specially benefit the chicken research community, directly contributing to the quality of the scientific research using the chicken model organism. At the time of this publication (February 2021), the zenodo tutorial (DOI:10.5281/zenodo.3765944) describing this package had been downloaded over 1820 times (in less than one year), showing that our package has attracted the attention of our target audience.

References

All publications are cited in line and hyperlinked to the original sources (with corresponding doi identifiers shown).

Funding

This scientific work was funded by FCT, Portugal (grant **PTDC/BEX-BID/5410/2014**) and Research Center Grant **UID/BIM/04773/2013 CBMR 1334**.

Author Contributions

ML collected the data, conducted the analysis, and wrote the manuscript. RM supervised the analysis, provided technical help, and wrote the manuscript. RPA devised the idea for the project, helped with interpreting the results, and wrote the manuscript. ID designed and supervised the analysis, interpreted the results, and wrote the paper. ***ID and ML contributed equally to this work (ordered alphabetically).** All authors read and approved the final manuscript.

Acknowledgements

The authors would like to thank the members of the Temporal Control of Cell Differentiation Lab for precious feedback and insightful scientific discussions.

*Isabel Duarte**
Centre for Biomedical Research (CBMR)
Universidade do Algarve, Faro, Portugal
Algarve Biomedical Center (ABC)
Universidade do Algarve, Faro, Portugal
**Equal Contribution*
ORCiD: 0000-0003-0060-2936
giduarte@ualg.pt

*Marta Liber**
Centre for Biomedical Research (CBMR)
Universidade do Algarve, Faro, Portugal
Algarve Biomedical Center (ABC)
Universidade do Algarve, Faro, Portugal
**Equal Contribution*
ORCiD: 0000-0003-4448-1937
mliber.pt@gmail.com

Ramiro Magno
Centre for Biomedical Research (CBMR)
Universidade do Algarve, Faro, Portugal
Algarve Biomedical Center (ABC)
Universidade do Algarve, Faro, Portugal

ORCiD: 0000-0001-5226-3441
ramiro.magno@gmail.com

Raquel P. Andrade
Centre for Biomedical Research (CBMR) | Algarve Biomedical Center (ABC)
Universidade do Algarve, Faro, Portugal
Department of Medicine and Biomedical Sciences (DCBM)
Universidade do Algarve, Faro, Portugal

ORCiD: 0000-0002-0397-5917
rgandrade@ualg.pt

Annex 3.2 - Publication of the "FrozenChicken" RData package in the *Zenodo* journal

April 25, 2020

Software | Open Access

FrozenChicken: Promoting the meta-analysis of chicken microarray data

(https://orcid.org/0000-0003-0060-2936) * Duarte, Isabel; (https://orcid.org/0000-0003-4448-1937) * Liber, Marta; (https://orcid.org/0000-0002-0397-5917) Andrade, Raquel P.

Summary

Background | *Gallus gallus* (chicken) is one of the most valuable model organisms for the study of the vertebrate embryo development. Such studies can be aided by pooling together OMICs (https://en.wikipedia.org/wiki/Omics) data from public repositories, like GEO (Gene Expression Omnibus (https://www.ncbi.nlm.nih.gov/geo/)) and ArrayExpress (https://www.ebi.ac.uk/arrayexpress/), that currently contain more than 11.660 datasets from chicken, representing a wealth of data that can be explored to answer fundamental questions and generate new hypotheses. However, since these data come from different experiments, their meta-analysis requires proper normalization to deal with the technical biases and batch effects before making the data comparable for statistical analysis.

Approach | An effective method for such normalization is the single-array pre-processing provided by the Frozen Robust Multiarray Analysis (https://pubmed.ncbi.nlm.nih.gov/20097884/) (**fRMA**) (McCall MN, et al. Biostatistics. 2010). This method uses *"frozen" RMA* vectors pre-computed from great amounts of available data for the same microarray platform, accounting for the aforementioned biases. However, such frozen vectors/parameters are available for multiple organisms (most notably, human, mouse, zebrafish, and fruit-fly among others), but not for chicken, hence preventing, or delaying, the proper meta-analysis of chicken transcriptomics datasets without prior computation of self-generated frozen parameters.

Output | Here, we present the RData package **FrozenChicken** containing the chicken microarray frozen vectors that can be directly plugged-in to a chicken microarray analysis pipeline (using fRMA) without any other prior data gathering and processing. The package is freely available for the research community at: https://github.com/iduarte/FrozenChicken (https://github.com/iduarte/FrozenChicken).

Significance | This package will directly benefit the chicken research community by facilitating future meta-analysis studies using transcriptomics datasets from public repositories, hence directly contributing to the quality of the scientific research using the chick model organism.

This scientific work was funded by FCT, Portugal (grant PTDC/BEX-BID/5410/2014) and Research Center Grant UID/BIM/04773/2013 CBMR 1334. | * These authors contributed equally to this work (ordered alphabetically).

Annex 4 | Tables of summary statistics of the datasets

Annex 4.1 - Table of descriptive statistics of the original PSM dataset, separated by the data origin

			Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
PSM - Initial dataset (n=29 886)	Array Express	Right 1	2.548	4.153	5.203	5.804	7.195	13.610
		Right 2	2.560	4.150	5.233	5.827	7.188	13.742
		Right 3	2.532	4.154	5.234	5.856	7.254	14.179
		Right 4	2.549	4.142	5.205	5.819	7.200	13.659
		Right 5	2.548	4.156	5.230	5.875	7.289	13.733
		Right 6	2.508	4.150	5.210	5.852	7.242	13.796
		Right 7	2.528	4.153	5.227	5.861	7.250	13.734
		Right 8	2.578	4.169	5.255	5.901	7.263	14.300
		Right 9	2.543	4.161	5.235	5.902	7.316	14.126
		Right 10	2.491	4.164	5.254	5.858	7.233	13.822
		Right 11	2.597	4.167	5.247	5.912	7.297	14.627
		Right 12	2.564	4.159	5.238	5.906	7.304	14.497
		Right 13	2.537	4.161	5.231	5.862	7.188	14.033
		Right 14	2.601	4.137	5.220	5.851	7.254	13.639
		Right 15	2.566	4.159	5.223	5.866	7.257	13.907
		Right 16	2.558	4.160	5.222	5.835	7.159	13.743
		Right 17	2.530	4.158	5.276	5.897	7.342	13.965
		Right 18	2.533	4.169	5.281	5.894	7.283	14.022
	GEO	Left 1	2.689	4.157	5.217	5.968	7.457	14.617
		Right 2	2.559	4.172	5.235	5.948	7.355	14.656
		Left 3	2.629	4.184	5.230	5.948	7.315	14.653
		Right 4	2.619	4.170	5.246	5.961	7.410	14.653
		Left 5	2.606	4.169	5.230	5.958	7.397	14.658
		Right 6	2.687	4.169	5.232	5.961	7.398	14.651
		Left 7	2.724	4.180	5.247	5.957	7.361	14.659
		Right 8	2.718	4.185	5.230	5.956	7.372	14.650
		Left 9	2.741	4.211	5.247	5.948	7.282	14.659
		Left 10	2.653	4.172	5.216	5.958	7.392	14.509
		Right 11	2.652	4.181	5.228	5.964	7.397	14.605
		Left 12	2.632	4.192	5.238	5.950	7.326	14.642
		Right 13	2.635	4.164	5.215	5.957	7.411	14.622
		Left 14	2.649	4.172	5.218	5.949	7.414	14.652

Annex 4.2 - Table of descriptive statistics of the original Limb dataset, separated by the data origin

			Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
Limb - Initial dataset (n=29 886)	Anterior	1	2.541	4.170	5.421	6.067	7.696	14.409
		2	2.548	4.182	5.398	6.045	7.623	14.514
		3	2.571	4.185	5.419	6.053	7.640	14.526
		4	2.582	4.156	5.407	6.043	7.623	14.481
		5	2.552	4.164	5.477	6.095	7.758	14.475
		6	2.548	4.137	5.475	6.100	7.789	14.438
		7	2.594	4.156	5.480	6.100	7.782	14.481
		8	2.600	4.134	5.411	6.074	7.767	14.391
		9	2.573	4.139	5.381	6.064	7.752	14.456
	Posterior	1	2.580	4.177	5.408	6.063	7.675	14.534
		2	2.537	4.161	5.424	6.060	7.681	14.395
		3	2.517	4.186	5.441	6.065	7.653	14.435
		4	2.530	4.140	5.465	6.103	7.802	14.429
		5	2.504	4.126	5.366	6.074	7.782	14.414

Annex 4.3 - Table of descriptive statistics of the intermediate PSM dataset containing only HVGs and separated by the data origin

			Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
PSM – Intermediate dataset of HVGs (n=9 527)	Array Express	Right 1	3.078	5.104	6.267	6.574	7.849	13.610
		Right 2	3.007	5.191	6.323	6.639	7.858	13.651
		Right 3	2.915	5.213	6.398	6.694	7.944	13.521
		Right 4	3.195	5.192	6.378	6.647	7.869	13.659
		Right 5	2.934	5.211	6.433	6.724	7.984	13.450
		Right 6	3.023	5.183	6.387	6.685	7.953	13.558
		Right 7	3.040	5.186	6.396	6.685	7.945	13.560
		Right 8	2.997	5.175	6.388	6.707	7.994	13.409
		Right 9	2.935	5.230	6.466	6.767	8.034	13.330
		Right 10	2.891	5.182	6.339	6.658	7.909	13.499
		Right 11	3.007	5.211	6.417	6.749	8.043	13.548
		Right 12	3.052	5.173	6.394	6.741	8.062	13.480
		Right 13	2.905	5.087	6.256	6.599	7.837	13.480
		Right 14	3.032	5.210	6.385	6.687	7.935	13.639
		Right 15	3.006	5.153	6.329	6.670	7.966	13.553
		Right 16	3.209	5.134	6.271	6.606	7.850	13.602
		Right 17	2.836	5.198	6.437	6.724	8.015	13.359
		Right 18	2.880	5.185	6.400	6.705	7.982	13.399
	GEO	Left 1	2.921	5.296	6.591	6.917	8.271	13.957
		Right 2	2.893	5.155	6.389	6.781	8.092	14.021
		Left 3	3.098	5.019	6.253	6.696	8.061	14.136
		Right 4	3.118	5.248	6.505	6.859	8.177	14.010
		Left 5	3.195	5.190	6.492	6.844	8.214	13.982
		Right 6	3.304	5.203	6.500	6.851	8.177	13.977
		Left 7	3.340	5.135	6.371	6.776	8.136	14.013
		Right 8	3.088	5.162	6.432	6.799	8.138	13.912
		Left 9	3.062	5.086	6.306	6.717	8.048	13.871
		Left 10	2.979	5.153	6.417	6.806	8.197	13.993
		Right 11	3.033	5.183	6.471	6.834	8.187	14.031
		Left 12	3.204	5.020	6.274	6.706	8.099	14.000
		Right 13	3.252	5.207	6.503	6.860	8.225	13.976
		Left 14	3.294	5.143	6.405	6.808	8.183	13.998

Annex 4.4 - Table of descriptive statistics of the intermediate Limb dataset containing only HVGs and separated by the data origin

			Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
Limb - intermediate dataset of HVGs ((n=10	Anterior	1	3.335	5.485	6.703	6.934	8.170	13.918
		2	3.419	5.406	6.621	6.862	8.091	13.822
		3	3.179	5.430	6.639	6.873	8.102	13.807
		4	3.205	5.450	6.664	6.901	8.126	13.875
		5	2.892	5.651	6.915	7.106	8.325	13.889
		6	3.232	5.730	6.948	7.142	8.364	13.789
		7	3.144	5.651	6.934	7.108	8.353	13.824
		8	3.286	5.554	6.817	7.033	8.320	13.779
		9	3.116	5.468	6.746	6.978	8.289	13.559
	Posterior	1	3.247	5.459	6.665	6.915	8.159	13.906
		2	3.204	5.482	6.702	6.936	8.189	13.942
		3	3.266	5.453	6.649	6.896	8.152	13.898
		4	3.005	5.695	6.919	7.122	8.341	13.664
		5	3.294	5.522	6.804	6.997	8.292	13.198

Annex 4.5 - Table of descriptive statistics of the final dataset of probe-sets comprehended in the PSM K1, separated by the data origin

			Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
PSM - Cluster 1 (n=128)	Array Express	Right 1	3.496	4.901	5.741	6.068	7.108	10.529
		Right 2	3.065	4.614	5.609	5.910	7.011	10.331
		Right 3	2.915	4.786	5.674	6.045	7.256	10.627
		Right 4	3.370	4.827	5.792	6.085	7.269	10.873
		Right 5	3.143	4.809	5.802	6.083	7.032	11.192
		Right 6	3.255	4.598	5.565	5.952	6.998	10.914
		Right 7	3.413	4.607	5.465	5.942	6.974	10.730
		Right 8	3.502	4.742	5.716	6.084	7.180	11.290
		Right 9	3.118	4.693	5.523	5.978	7.038	10.742
		Right 10	3.275	4.704	5.840	6.055	7.100	10.461
		Right 11	3.438	4.900	6.005	6.265	7.489	11.121
		Right 12	3.254	4.734	5.636	6.089	7.170	11.172
		Right 13	3.387	4.832	5.550	6.082	7.324	11.094
		Right 14	3.281	4.757	5.662	5.987	7.097	10.481
		Right 15	3.642	4.882	5.850	6.199	7.303	11.285
		Right 16	3.268	4.788	5.653	5.983	6.982	10.268
		Right 17	3.049	4.623	5.695	6.021	7.259	10.873
		Right 18	2.880	4.725	5.611	6.028	7.131	11.070
	GEO	Left 1	3.403	4.756	5.574	6.054	7.199	10.920
		Right 2	3.329	4.812	5.511	6.029	7.028	10.192
		Left 3	3.612	4.968	5.619	6.129	7.152	10.681
		Right 4	3.133	4.776	5.622	6.091	7.209	10.587
		Left 5	3.763	4.941	6.155	6.374	7.730	11.800
		Right 6	3.304	4.906	5.794	6.274	7.579	11.531
		Left 7	3.639	5.126	6.386	6.655	8.062	12.149
		Right 8	3.219	4.944	6.011	6.406	7.866	12.075
		Left 9	3.776	5.450	6.834	7.159	8.771	12.551
		Left 10	4.205	6.554	8.012	8.055	9.354	12.729
		Right 11	3.767	6.167	7.794	7.766	9.216	12.407
		Left 12	5.473	7.463	9.053	8.943	10.033	12.938
		Right 13	5.062	7.852	8.916	9.024	10.078	12.822
		Left 14	5.022	7.839	9.130	9.188	10.292	13.413

Annex 4.6 - Table of descriptive statistics of the final dataset of probe-sets comprehended in the PSM K2, separated by the data origin

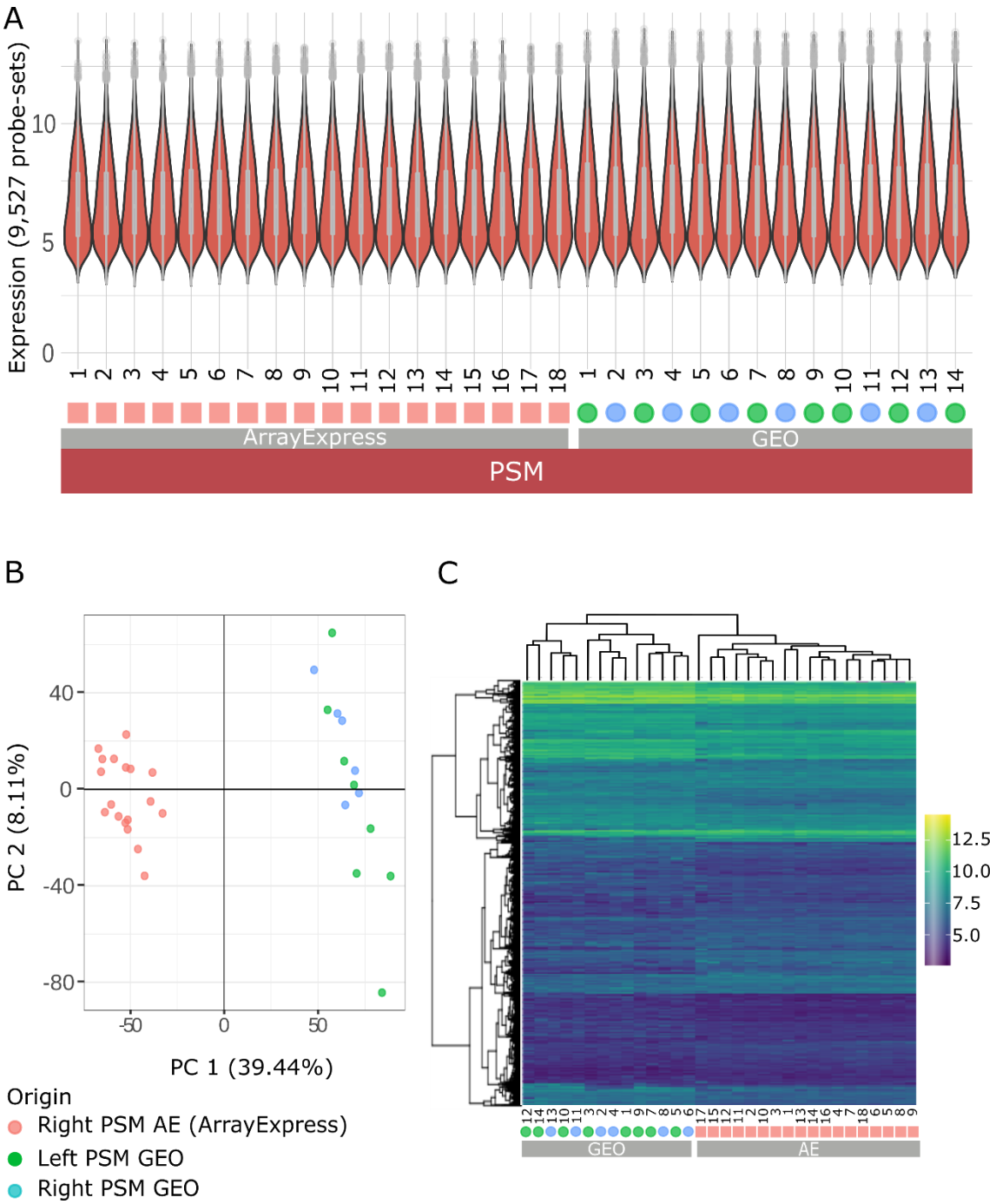
			Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
PSM - Cluster 2 (n=37)	Array Express	Right 1	3.296	4.354	6.280	6.086	6.838	10.636
		Right 2	3.254	4.398	5.513	5.953	6.764	11.667
		Right 3	3.350	4.505	6.471	6.389	7.300	11.431
		Right 4	3.386	4.156	5.570	5.779	6.768	11.472
		Right 5	3.366	4.502	5.722	6.383	8.070	11.705
		Right 6	3.268	4.224	6.120	5.953	6.605	11.652
		Right 7	3.563	4.091	5.394	5.817	6.677	11.743
		Right 8	3.404	4.172	5.777	6.093	7.186	11.500
		Right 9	3.488	4.806	5.889	6.577	7.614	11.725
		Right 10	3.421	4.751	6.627	6.608	7.786	11.210
		Right 11	3.311	4.612	6.285	6.431	7.203	11.480
		Right 12	3.248	4.296	5.651	5.952	6.820	12.387
		Right 13	3.343	4.454	5.477	5.977	7.188	11.301
		Right 14	3.338	4.703	5.916	6.059	7.423	11.183
		Right 15	3.487	4.370	6.048	6.339	8.199	12.013
		Right 16	3.469	4.402	5.939	6.236	7.915	11.374
		Right 17	3.391	5.539	7.038	7.232	8.705	11.769
		Right 18	3.652	4.589	6.103	6.451	7.801	11.975
	GEO	Left 1	3.778	6.239	7.489	7.713	9.178	11.929
		Right 2	4.250	5.951	6.834	7.429	8.760	12.003
		Left 3	3.371	5.364	6.604	6.719	8.101	11.306
		Right 4	3.488	5.367	6.584	6.651	7.705	11.452
		Left 5	3.505	4.503	5.902	6.283	7.596	11.501
		Right 6	3.397	4.500	5.947	6.441	8.085	11.122
		Left 7	3.358	4.304	5.849	6.206	7.411	10.958
		Right 8	3.088	4.396	5.898	6.328	7.986	11.059
		Left 9	3.062	4.403	5.831	6.291	7.658	11.512
		Left 10	3.479	4.653	6.249	6.545	8.363	10.932
		Right 11	3.147	4.823	5.901	6.424	7.682	11.136
		Left 12	3.395	4.443	6.052	6.251	7.832	11.283
		Right 13	3.252	4.322	5.856	6.395	7.525	11.477
		Left 14	3.484	4.204	5.768	6.166	7.491	10.642

Annex 4.7 - Table of descriptive statistics of the final dataset of probe-sets comprehended in the Limb K1, separated by the data origin

			Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
Limb - Cluster 1 (n=173)	Anterior	1	3.441	5.358	6.637	6.693	7.645	11.726
		2	3.419	5.145	6.457	6.531	7.576	11.586
		3	3.421	5.249	6.526	6.568	7.594	11.534
		4	3.275	5.365	6.573	6.639	7.609	11.459
		5	3.597	6.108	7.395	7.381	8.377	11.358
		6	3.271	6.271	7.462	7.381	8.339	11.543
		7	3.312	6.154	7.379	7.328	8.352	11.577
		8	3.501	5.633	7.004	6.968	7.964	11.582
		9	3.552	5.372	6.553	6.755	7.925	11.773
	Posterior	1	3.697	5.555	6.645	6.743	7.732	12.681
		2	3.733	5.597	6.807	6.801	7.813	12.647
		3	3.571	5.515	6.828	6.782	7.730	12.700
		4	4.001	6.482	7.434	7.458	8.342	12.500
		5	3.731	5.518	6.754	6.821	7.834	12.613

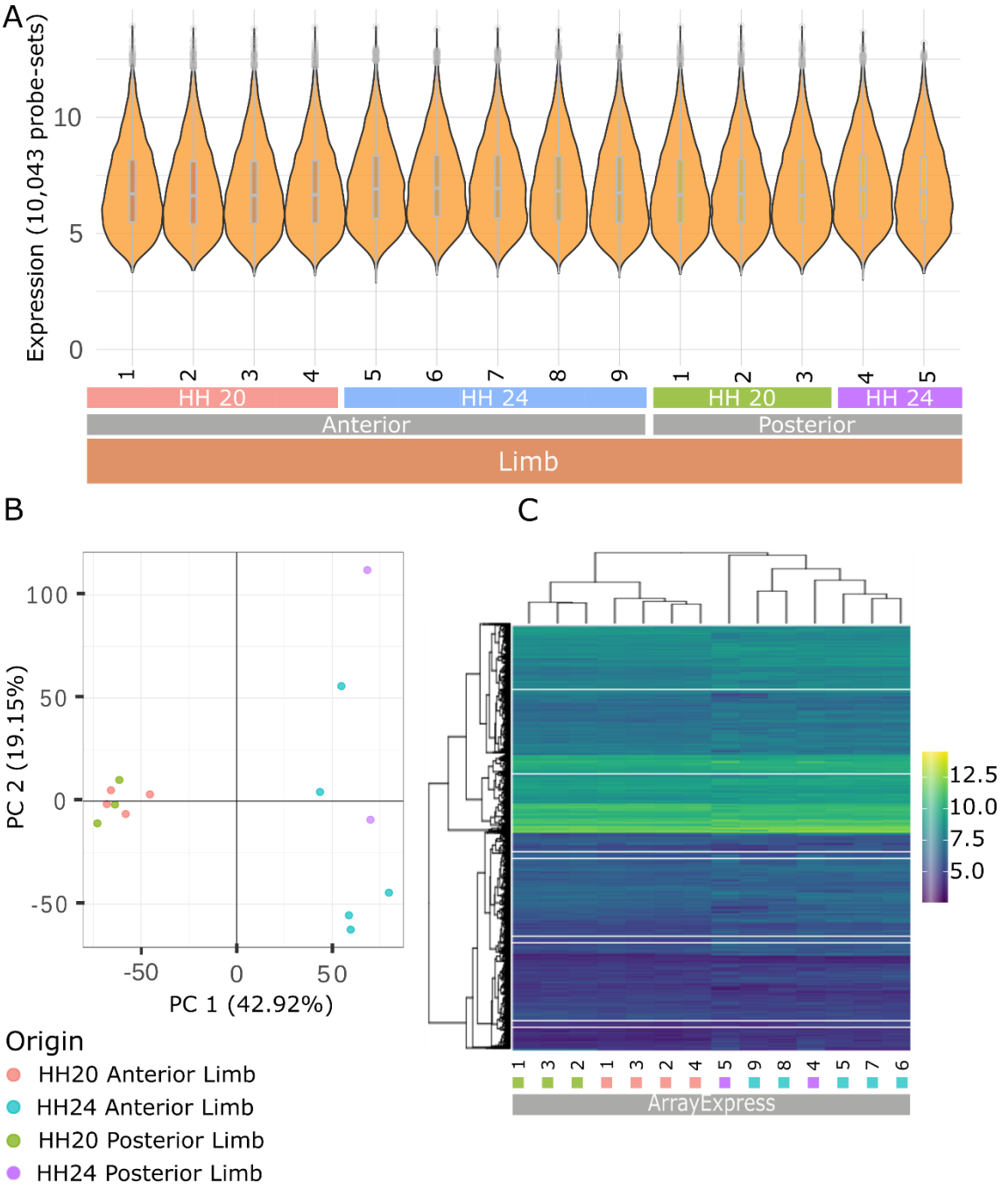
Annex 5 | Graphical representation of the summary statistics of the datasets

Annex 5.1 - Quality Control statistics for the PSM intermediate dataset of HVGs



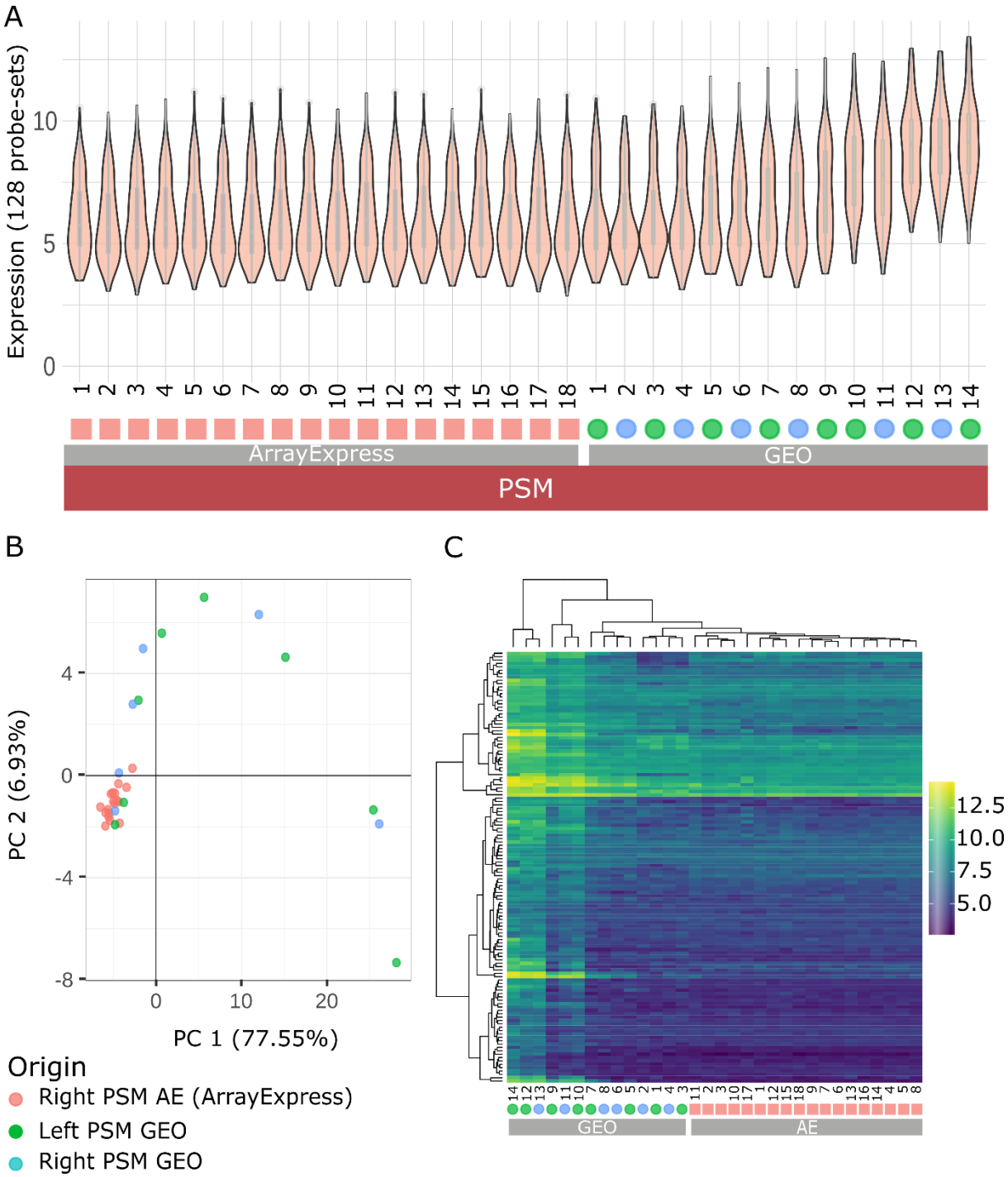
Quality assessment of the Intermediate dataset of the PSM HVGs. A | Comparison of the distribution of log2-transformed gene expression values from an intermediate dataset of PSM HVGs. Each violin plot corresponds to one array, and the outline in black shows kernel probability density, i.e. the width of the colored area corresponds to the proportion of the data points located there. The inner boxplot indicates the median and IQR. Sample order has no meaning. **B |** PCA plot showing the cross-array variability across the 2 principal components. The x-axis corresponds to the first component and the y axis represents the second component. **C |** Heatmap visualization of the expression values. Lateral clustering is performed across genes. Sample clustering is displayed on top. Low gene expression is coded by darker colors, whereas high expressions are color-coded in lighter shades. Samples = 32; HVGs = Highly Variable Genes.

Annex 5.2 - Quality Control statistics for the Limb intermediate dataset of HVGs



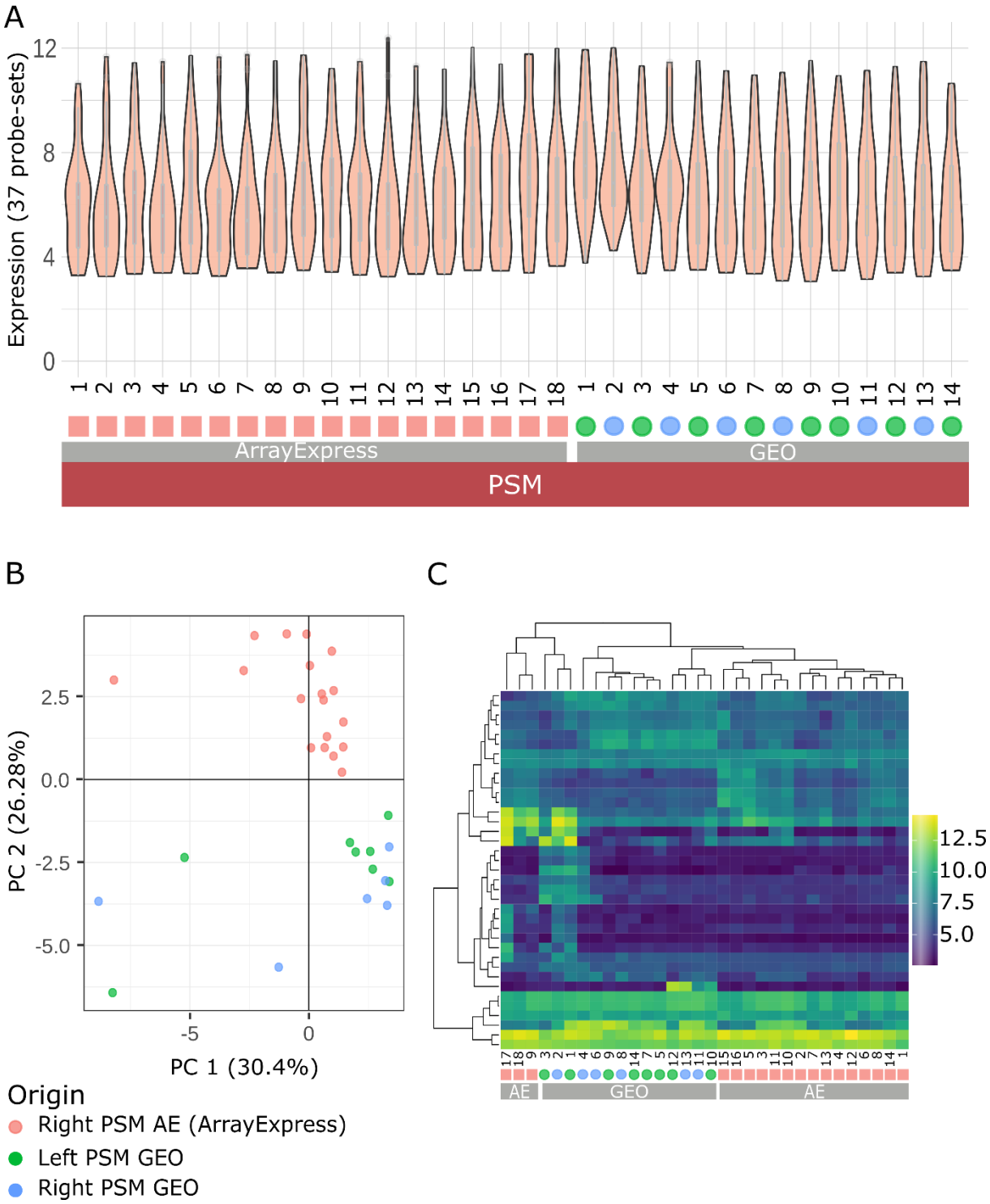
Quality assessment of the Intermediate dataset of the Limb HVGs. A | Comparison of the distribution of log2-transformed gene expression values from an intermediate dataset of limb HVGs. Each violin plot corresponds to one array, and the outline in black shows kernel probability density, i.e. the width of the colored area corresponds to the proportion of the data points located there. The inner boxplot indicates the median and IQR. Sample order has no meaning. **B |** PCA plot showing the cross-array variability across the 2 principal components. The x-axis corresponds to the first component and the y axis represents the second component. **C |** Heatmap visualization of the expression values. Lateral clustering is performed across genes. Sample clustering is displayed on top. Low gene expression is coded by darker colors, whereas high expressions are color-coded in lighter shades. Samples = 14; HVGs = Highly Variable Genes.

Annex 5.3 - Quality Control statistics for the PSM K1



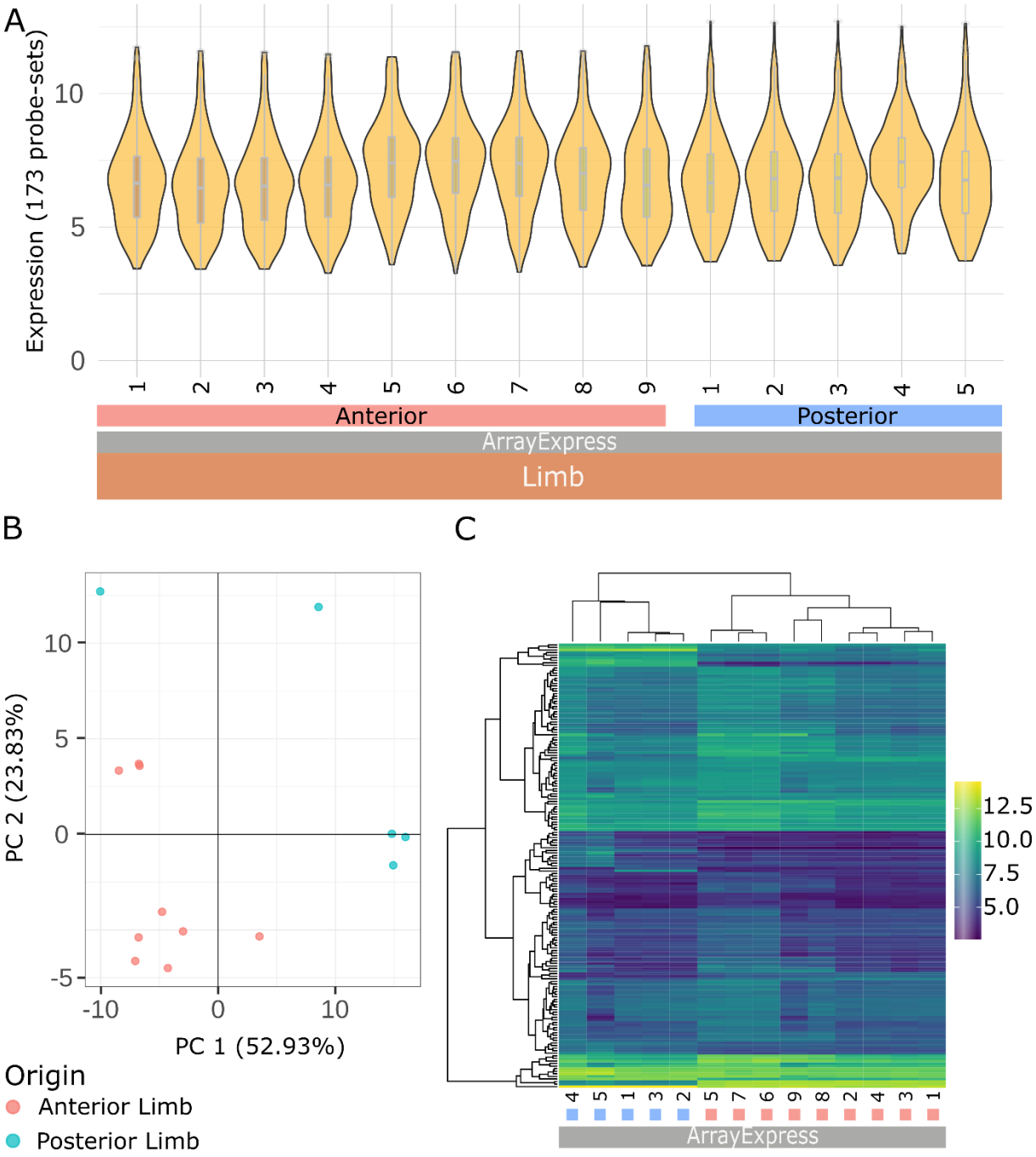
Quality assessment of the PSM K1. A | Comparison of the distribution of log2-transformed gene expression values from PSM K1. Each violin plot corresponds to one array, and the outline in black shows kernel probability density, i.e. the width of the colored area corresponds to the proportion of the data points located there. The inner boxplot indicates the median and IQR. Sample order has no meaning. **B |** PCA plot showing the cross-array variability across the 2 principal components. The x-axis corresponds to the first component and the y axis represents the second component. **C |** Heatmap visualization of the expression values. Lateral clustering is performed across genes. Sample clustering is displayed on top. Low gene expression is coded by darker colors, whereas high expressions are color-coded in lighter shades. Samples = 32.

Annex 5.4 - Quality Control statistics for the PSM K2



Quality assessment of the PSM K2. A | Comparison of the distribution of log2-transformed gene expression values from PSM K2. Each violin plot corresponds to one array, and the outline in black shows kernel probability density, i.e. the width of the colored area corresponds to the proportion of the data points located there. The inner boxplot indicates the median and IQR. Sample order has no meaning. **B |** PCA plot showing the cross-array variability across the 2 principal components. The x-axis corresponds to the first component and the y axis represents the second component. **C |** Heatmap visualization of the expression values. Lateral clustering is performed across genes. Sample clustering is displayed on top. Low gene expression is coded by darker colors, whereas high expressions are color-coded in lighter shades. Samples = 32.

Annex 5.5 - Quality Control statistics for the Limb K1



Quality assessment of the Limb K1. A | Comparison of the distribution of log2-transformed gene expression values from Limb K1. Each violin plot corresponds to one array, and the outline in black shows kernel probability density, i.e. the width of the colored area corresponds to the proportion of the data points located there. The inner boxplot indicates the median and IQR. Sample order has no meaning. **B |** PCA plot showing the cross-array variability across the 2 principal components. The x-axis corresponds to the first component and the y axis represents the second component. **C |** Heatmap visualization of the expression values. Lateral clustering is performed across genes. Sample clustering is displayed on top. Low gene expression is coded by darker colors, whereas high expressions are color-coded in lighter shades. Samples = 14.

Annex 6 | Complete ClockOME list of 296 oscillatory genes retrieved by Oscope, separated by cluster

Annex 6.1 - List of genes in the PSM K1

PSM Cluster 1 (106 genes / 128 probes)				
	Probe ID	ENSEMBL	SYMBOL	Enrichment Status
1	Gga.10980.2.S1_at	ENSGALG00000007079	PAPPA	
2	Gga.11430.1.S1_at	ENSGALG00000015419	PENK	
3	Gga.1150.2.S1_a_at	ENSGALG00000001956	OLFML3	BP, CC
4	Gga.120.1.S1_a_at	ENSGALG00000004093	TBX22	
5	Gga.123.1.S1_at	ENSGALG00000015728	MUSK	BP, MF, CC
6	Gga.12367.1.S1_at	ENSGALG00000013006	ROPN1L	
7	Gga.12614.1.S1_at	ENSGALG00000007174	TNFSF15	
8	Gga.12811.1.S1_at	ENSGALG00000036616	NUAK2	
9	Gga.12987.1.S1_at	ENSGALG00000037525	FGFRL1	BP, MF, CC
10	Gga.8771.1.S1_at			
11	Gga.13008.1.S1_at	ENSGALG00000004530	MMD2	
12	Gga.13096.1.S1_at	ENSGALG00000035845	JPH1	
13	Gga.13162.1.S1_at	ENSGALG00000015590	MANEA	
14	Gga.135.3.S1_a_at	ENSGALG00000015422	NRG1	BP, MF, CC
15	Gga.13574.1.S1_at	ENSGALG00000031487	FSTL4	
16	Gga.8081.1.S1_at			
17	Gga.1363.1.S1_at	ENSGALG00000014508	CD38	
18	Gga.13651.1.S1_at	ENSGALG00000043204	PAX7	
19	Gga.555.1.S1_at			
20	Gga.1479.2.S1_a_at	ENSGALG00000040493	PTN	BP, MF, CC
21	Gga.15222.1.S1_at	ENSGALG00000013598	SCGN	
22	Gga.15370.1.S1_at	ENSGALG00000027198	TMEM47	
23	Gga.17453.1.S1_at			
24	Gga.15420.1.S1_at	ENSGALG00000037506	SIPA1L2	
25	Gga.20062.1.S1_s_at			
26	Gga.15757.1.S1_at	ENSGALG00000029102	PXYLP1	
27	Gga.16002.1.S1_at	ENSGALG00000016967	ENOX1	
28	Gga.16244.1.S1_a_at	ENSGALG00000040866	P3H2	BP, MF, CC
29	Gga.16413.1.A1_a_at	ENSGALG00000015708	FGFR3	BP, MF, CC
30	Gga.16413.3.S1_at			
31	Gga.14066.1.S1_at			
32	Gga.16444.1.S1_at	ENSGALG00000009700	PDK4	
33	Gga.16813.1.S1_at	ENSGALG00000031929	KCNMB2	
34	Gga.16845.1.S1_at	ENSGALG00000011003	SLC35F3	
35	Gga.17559.1.S1_at	ENSGALG00000012620	PTCH1	BP, MF, CC
36	Gga.17628.1.S1_at	ENSGALG00000032181	LRP4	
37	Gga.18227.1.S1_at	ENSGALG00000013583	FAM114A1	

38	Gga.1839.1.S1_at	ENSGALG00000004508	**EYA2**	
39	Gga.19162.1.S1_at	ENSGALG00000002555	**RET**	
40	Gga.19165.1.S1_at	ENSGALG00000032770	**VRK2**	
41	Gga.1944.1.S1_at	ENSGALG00000006783	**PLOD2**	
42	Gga.1969.1.S1_at	--	**LOC423523**	
43	Gga.198.1.S1_at	ENSGALG00000006992	**MMP9**	
44	Gga.20018.1.S1_at	ENSGALG00000006409	**PODXL**	BP, CC
45	Gga.2136.1.S1_a_at	ENSGALG00000025738	**RHOU**	
46	Gga.2136.1.S1_at			
47	Gga.2516.1.S1_at	ENSGALG00000007177	**PTPRG**	BP, MF, CC
48	Gga.253.1.S1_at	ENSGALG00000012644	**FOXD1**	BP, MF, CC
49	Gga.2606.1.A1_at	ENSGALG00000034346	**RDH10**	
50	Gga.2606.1.S1_at			
51	Gga.2606.1.S1_x_at			
52	Gga.2668.1.S1_a_at	ENSGALG00000011298	**RARB**	BP, MF, CC
53	Gga.276.1.S1_at	ENSGALG00000003699	**EBF1**	
54	Gga.288.1.S1_at	ENSGALG00000001191	**MEOX1**	
55	Gga.2996.1.S2_at	ENSGALG00000004270	**ALDH1A2**	BP, MF, CC
56	Gga.2996.2.S1_a_at			
57	Gga.3178.1.S2_at	ENSGALG00000006886	**DACH2**	
58	Gga.3204.1.S1_at	ENSGALG00000006633	**TMEM243**	
59	Gga.3264.1.S1_at	ENSGALG00000010836	**AHR**	
60	Gga.3279.1.S1_at	ENSGALG00000026276	**TCF15**	BP, MF, CC
61	Gga.3323.1.S1_s_at	ENSGALG00000041344	**FABP5**	
62	Gga.3330.1.S1_at	ENSGALG00000038775	**SPON1**	BP, MF, CC
63	Gga.3412.1.S1_at	ENSGALG00000000678	**CITED4**	BP, MF, CC
64	Gga.3615.1.S1_at	ENSGALG00000014908	**FST**	BP, MF, CC
65	Gga.3615.1.S2_at			
66	Gga.3665.1.S1_a_at	ENSGALG00000012906	**CDH20**	BP, MF, CC
67	Gga.3665.1.S2_at			
68	Gga.3745.1.S1_at	ENSGALG00000030209	**CPZ**	BP, MF, CC
69	Gga.3745.1.S2_at			
70	Gga.3973.1.S1_at	ENSGALG00000041640	**TWIST1**	
71	Gga.3986.2.S1_a_at	ENSGALG00000030781	**CA2**	BP, MF, CC
72	Gga.4309.1.S1_at	ENSGALG00000005263	**SOX8**	BP, MF, CC
73	Gga.4445.1.S1_at	ENSGALG00000007000	**NR2F2**	BP, MF, CC
74	Gga.4481.1.S1_at	ENSGALG00000008294	**ITPR1**	
75	Gga.4602.1.S1_at	ENSGALG00000003548	**BAIAP2L1**	
76	Gga.481.1.S1_at	ENSGALG00000034067	**FMOD**	CC
77	Gga.5073.1.S1_at	--	**PI3**	
78	Gga.5217.1.S1_at	ENSGALG00000010988	**MPP6**	
79	Gga.5225.1.S1_at	ENSGALG00000035419	**CDON**	
80	Gga.5316.1.S1_at	ENSGALG00000012505	**LRFN5**	
81	Gga.5316.2.S1_s_at			

82	Gga.5676.1.S1_at	ENSGALG00000035052	**BMP3**	
83	Gga.5847.1.S1_at	ENSGALG00000016001	**LZTS3**	
84	Gga.5933.1.S1_at	ENSGALG00000034456	**PRELP**	
85	Gga.6141.1.S1_at	ENSGALG00000037479	**IGSF21**	
86	Gga.6311.1.S1_at	ENSGALG00000035031	**HEY1**	
87	Gga.6433.1.S1_at	ENSGALG00000007311	**CLDN2**	
88	Gga.6877.1.S1_at			
89	GgaAffx.9882.1.S1_at	ENSGALG00000015519	**ROBO2**	
90	GgaAffx.9882.2.S1_s_at			
91	Gga.7194.1.S1_at			
92	GgaAffx.21208.1.S1_s_at	ENSGALG00000019077	**SPATA13**	
93	Gga.7235.1.S1_at	ENSGALG00000030376	**DHRS3**	
94	Gga.7458.1.S1_at	ENSGALG00000001968	**SYT6**	
95	Gga.7533.1.S1_at	ENSGALG00000043035	**SHISA2**	
96	Gga.779.1.S1_a_at	ENSGALG00000007284	**SYP**	
97	Gga.792.1.S1_at	ENSGALG00000029072	**NTN3**	BP, MF, CC
98	Gga.7965.1.S1_at	ENSGALG00000037246	**RAMP3**	
99	Gga.7979.1.S1_at	ENSGALG00000002818	**VAT1**	
100	Gga.7985.1.S1_at			
101	GgaAffx.3750.1.S1_at	ENSGALG00000027905	**CSRNP1**	
102	GgaAffx.21217.1.S1_at	ENSGALG00000002098	**GRIK3**	
103	Gga.805.1.S1_at	ENSGALG00000015403	**EPHA3**	BP, MF, CC
104	Gga.8360.1.S1_at	ENSGALG00000033083	**ATOH8**	
105	Gga.837.1.S1_a_at	ENSGALG00000010983	**NPY**	BP, MF, CC
106	Gga.90.1.S1_at	ENSGALG00000010794	**MEOX2**	
107	Gga.9295.1.S1_at	ENSGALG00000017304	**PGM2L1**	
108	Gga.9773.1.S1_s_at			
109	GgaAffx.11584.1.S1_s_at	ENSGALG00000012712	**RBM24**	BP, MF, CC
110	GgaAffx.12050.1.S1_s_at	ENSGALG00000038217	**CD82**	
111	GgaAffx.12393.1.S1_at			
112	GgaAffx.12393.1.S1_s_at	ENSGALG00000040031	**SLC35G2**	CC
113	GgaAffx.12879.1.S1_s_at	ENSGALG00000040023	**MAPK11**	
114	GgaAffx.13061.1.S1_at	ENSGALG00000002225	**RFFL**	
115	GgaAffx.21044.1.S1_at			
116	GgaAffx.21044.1.S1_s_at	ENSGALG00000011288	**DACT2**	
117	GgaAffx.23308.3.S1_s_at	ENSGALG00000040156	**CCDC88B**	
118	GgaAffx.23767.1.S1_at	ENSGALG00000037629	**TRANK1**	
119	GgaAffx.25640.1.S1_at	ENSGALG00000002804	**RND2**	
120	GgaAffx.25649.1.S1_at	ENSGALG00000002840	**GRM4**	
121	GgaAffx.3795.1.S1_at	ENSGALG00000006087	**GPC3**	
122	GgaAffx.4561.1.S1_at	--	**RIPPLY1**	
123	GgaAffx.4848.1.S1_s_at	ENSGALG00000030821	**CBLN3**	
124	GgaAffx.5227.1.S1_at	ENSGALG00000008275	**MESP2**	

125	GgaAffx.5697.2.S1_s_at	ENSGALG00000009017	**STEAP2**	
126	GgaAffx.6392.1.S1_at	ENSGALG00000010161	**DMRT3**	
127	GgaAffx.7250.1.S1_at	ENSGALG00000011414	**AMDHD1**	
128	GgaAffx.7674.1.S1_at	ENSGALG00000035505	**AOAH**	

Annex 6.2 - List of genes in the PSM K2

PSM Cluster 2 (32 genes / 37 probes)				
	Probe ID	**ENSEMBL**	**SYMBOL**	**Enrichment status**
1	Gga.10559.1.S1_at	ENSGALG00000002145	**CAMSAP2**	
2	Gga.10753.2.S1_a_at	ENSGALG00000002904	**REEP3**	
3	Gga.12040.2.S1_a_at	ENSGALG00000010152	**TSPAN8**	
4	Gga.12510.1.S1_at	--	**LOC420043**	
5	Gga.13903.1.S1_at	ENSGALG00000030237	**PITX1**	BP, MF, CC
6	Gga.15973.1.S1_at	ENSGALG00000040189	**RIPK4**	
7	Gga.16081.1.S1_at	--	**LOC769756**	
8	Gga.16552.1.S1_a_at	ENSGALG00000035599	**CD24**	
9	Gga.16760.1.S1_at	ENSGALG00000028471	**CNPY3**	
10	Gga.16760.1.S1_s_at			
11	Gga.1819.1.S1_at	ENSGALG00000026005	**DIO2**	BP, MF, CC
12	Gga.2413.1.S1_at	ENSGALG00000017347	**HBBR**	BP, MF, CC
13	Gga.2902.1.S1_s_at	ENSGALG00000031597	**HBAD**	BP, MF, CC
14	Gga.2909.1.S1_at	ENSGALG00000043234	**HBA1**	BP, MF, CC
15	Gga.3315.2.S1_a_at	ENSGALG00000002605	**MRPL17**	
16	Gga.3315.2.S1_x_at			
17	Gga.3398.1.S1_a_at	ENSGALG00000042492	**PITX2**	BP, MF, CC
18	Gga.3398.2.S1_a_at			
19	Gga.4439.2.S1_at	ENSGALG00000007441	**MAP6**	BP, MF, CC
20	GgaAffx.4641.1.S1_s_at			
21	Gga.4875.1.S1_at	ENSGALG00000023740	**HBZ**	BP, MF, CC
22	Gga.6976.1.S1_at	ENSGALG00000031496	**SPINK5**	BP, MF, CC
23	Gga.7231.1.S1_at	ENSGALG00000032994	**SOX17**	
24	Gga.7339.1.S1_at	ENSGALG00000013907	**KDR**	
25	Gga.7813.1.S1_at	ENSGALG00000012126	**CFI**	
26	Gga.9481.1.S1_s_at	ENSGALG00000034868	**KRT7**	CC
27	GgaAffx.10087.1.S1_at	ENSGALG00000015845	**RIPPLY2**	
28	GgaAffx.12062.1.S1_s_at	ENSGALG00000001161	**FLI1**	
29	GgaAffx.20857.1.S1_s_at	ENSGALG00000011814	**ATF7IP**	BP, CC
30	GgaAffx.21787.1.S1_s_at	ENSGALG00000040249	**BHLHE22**	BP, MF, CC
31	GgaAffx.2588.1.S1_at	ENSGALG00000004186	**NSRP1**	

32	GgaAffx.2588.1.S1_x_at			
33	GgaAffx.26068.1.S1_s_at	ENSGALG00000003947	**HNRNPH3**	
34	GgaAffx.3482.4.S1_at	ENSGALG00000005549	**CUEDC1**	
35	GgaAffx.3856.1.S1_s_at	ENSGALG00000006182	**NSFL1C**	BP, MF, CC
36	GgaAffx.4797.1.S1_s_at	ENSGALG00000007703	**LOC422320**	
37	GgaAffx.6790.1.S1_at	--	**LOC101749678**	

Annex 6.3 - List of genes in the Limb K1

Limb Cluster 1 (163 genes / 173 probes)				
	Probe ID	**ENSEMBL**	**SYMBOL**	**Enrichment Status**
1	Gga.10094.1.S1_s_at	ENSGALG00000040938	**NLGN1**	
2	Gga.10102.1.S1_s_at	ENSGALG00000002246	**DENND1B**	
3	Gga.10257.2.S1_a_at	ENSGALG00000000195	**SLC23A2**	
4	Gga.10446.1.S1_at	ENSGALG00000034803	**RNF151**	
5	Gga.10625.1.S1_at	ENSGALG00000014845	**PLCXD3**	
6	Gga.10737.1.S1_s_at	ENSGALG00000002750	**NCKAP1**	
7	Gga.10933.1.S1_at	ENSGALG00000013602	**PCM1**	BP, CC
8	Gga.10957.1.S1_s_at	ENSGALG00000034271	**UHRF1**	
9	Gga.11201.1.S1_s_at	ENSGALG00000008477	**EXOC4**	
10	Gga.11201.2.S1_s_at			
11	Gga.11258.1.S1_at	ENSGALG00000017125	**MICU2**	
12	Gga.11364.1.S1_s_at	ENSGALG00000002346	**CLSPN**	
13	Gga.11453.1.S1_at	ENSGALG00000004710	**TMOD3**	
14	Gga.11463.2.S1_s_at	ENSGALG00000032746	**ENPP2**	
15	Gga.12157.1.S1_at	ENSGALG00000042511	**PKDCCA**	
16	Gga.12327.1.S1_at	ENSGALG00000011357	**VEZT**	
17	Gga.12349.1.S1_at	ENSGALG00000006507	**CLMP**	
18	Gga.12376.1.S1_at	ENSGALG00000011251	**TBC1D5**	
19	Gga.12380.1.S1_s_at	ENSGALG00000023451	**TRPM7**	
20	Gga.12449.1.S1_s_at	ENSGALG00000036365	**POGZ**	
21	Gga.12539.2.S1_at	ENSGALG00000004583	**NPLOC4**	
22	Gga.12980.1.S1_s_at	ENSGALG00000005805	**PLCD1**	
23	Gga.13492.1.S1_s_at	ENSGALG00000010314	**NEK9**	
24	Gga.13574.1.S1_at	ENSGALG00000031487	**FSTL4**	
25	Gga.13581.1.S1_s_at	ENSGALG00000015763	**GABPA**	
26	Gga.13591.1.S1_s_at	ENSGALG00000030342	**SARNP**	
27	Gga.1363.1.S1_at	ENSGALG00000014508	**CD38**	
28	Gga.13845.1.S1_at	ENSGALG00000014831	**TRMT11**	BP, MF, CC
29	Gga.14338.1.S1_at	ENSGALG00000009495	**FGFR2**	BP, MF, CC

30	Gga.14418.1.S1_s_at	ENSGALG00000006571	**SETD5**	
31	Gga.14494.1.S1_at	ENSGALG00000038848	**MSX2**	BP, MF, CC
32	Gga.1473.1.S1_at	ENSGALG00000010870	**JUN**	BP, MF, CC
33	Gga.14861.1.S1_s_at	ENSGALG00000004949	**KIAA0355**	
34	Gga.15131.1.S1_at	ENSGALG00000003875	**ABL1**	
35	Gga.15365.1.S1_s_at	ENSGALG00000008432	**AP2M1**	BP, MF, CC
36	Gga.1585.1.S2_at	ENSGALG00000039690	**STMN2**	BP, MF, CC
37	Gga.16081.1.S1_at	--	**LOC769756**	
38	Gga.16226.1.S1_s_at	ENSGALG00000002942	**JMJD1C**	
39	Gga.16294.1.S1_a_at	ENSGALG00000036300	**SEC61A2**	
40	Gga.16552.1.S1_a_at	ENSGALG00000035599	**CD24**	
41	Gga.16552.2.S1_a_at			
42	Gga.1669.1.S1_a_at	ENSGALG00000036971	**FGF12**	
43	Gga.16855.1.S1_at	ENSGALG00000037791	**FRMD4B**	
44	Gga.16883.1.S1_at	--	**LONRF2**	
45	Gga.16949.1.S1_s_at	ENSGALG00000031917	**ANKIB1**	
46	Gga.17077.1.S1_s_at	ENSGALG00000011811	**SYNE2**	
47	Gga.17455.1.S1_at	ENSGALG00000002223	**LHX9**	BP, MF, CC
48	Gga.2348.2.S1_at			
49	Gga.17508.1.S2_at	ENSGALG00000014970	**FSTL1**	
50	Gga.17559.1.S1_at	ENSGALG00000012620	**PTCH1**	BP, MF, CC
51	Gga.17645.1.S1_s_at	ENSGALG00000002929	**TRIP12**	
52	Gga.1784.1.S1_at	ENSGALG00000008747	**ITGA8**	BP, MF, CC
53	Gga.18148.1.S1_at	ENSGALG00000050615	**CKAP2L**	
54	Gga.1817.1.S1_at	ENSGALG00000039238	**SALL1**	
55	Gga.18569.1.S1_s_at	ENSGALG00000008616	**DDR1**	
56	Gga.18916.1.S1_at	ENSGALG00000002922	**NOTCH2**	
57	Gga.19526.1.S1_at	ENSGALG00000042557	**LOC421238**	
58	Gga.19791.1.S1_s_at	ENSGALG00000033630	**LRP12**	
59	Gga.19872.1.S1_s_at	ENSGALG00000008625	**ANAPC2**	
60	Gga.19878.1.S1_at	ENSGALG00000029316	**ARID1A**	
61	Gga.20021.1.S1_s_at	ENSGALG00000032685	**HECTD4**	
62	Gga.17473.1.S1_s_at	ENSGALG00000013757	**TMX3**	
63	Gga.261.3.S1_a_at	ENSGALG00000003446	**PRLR**	BP, MF, CC
64	Gga.2685.1.S2_at	ENSGALG00000006508	**FGF13**	
65	Gga.2758.1.S1_at	ENSGALG00000003589	**VTN**	
66	Gga.2931.1.S1_at	ENSGALG00000008676	**RRBP1**	
67	Gga.3093.1.S1_at	ENSGALG00000038154	**YAP1**	BP, MF, CC
68	Gga.3219.1.S1_at	ENSGALG00000016558	**VEGFD**	
69	Gga.3259.1.S1_at	ENSGALG00000036189	**ANGPT2**	
70	Gga.3363.2.S1_a_at	ENSGALG00000010642	**IRF2**	BP, MF, CC
71	Gga.3950.1.S1_at	ENSGALG00000029301	**BMP2**	BP, MF, CC

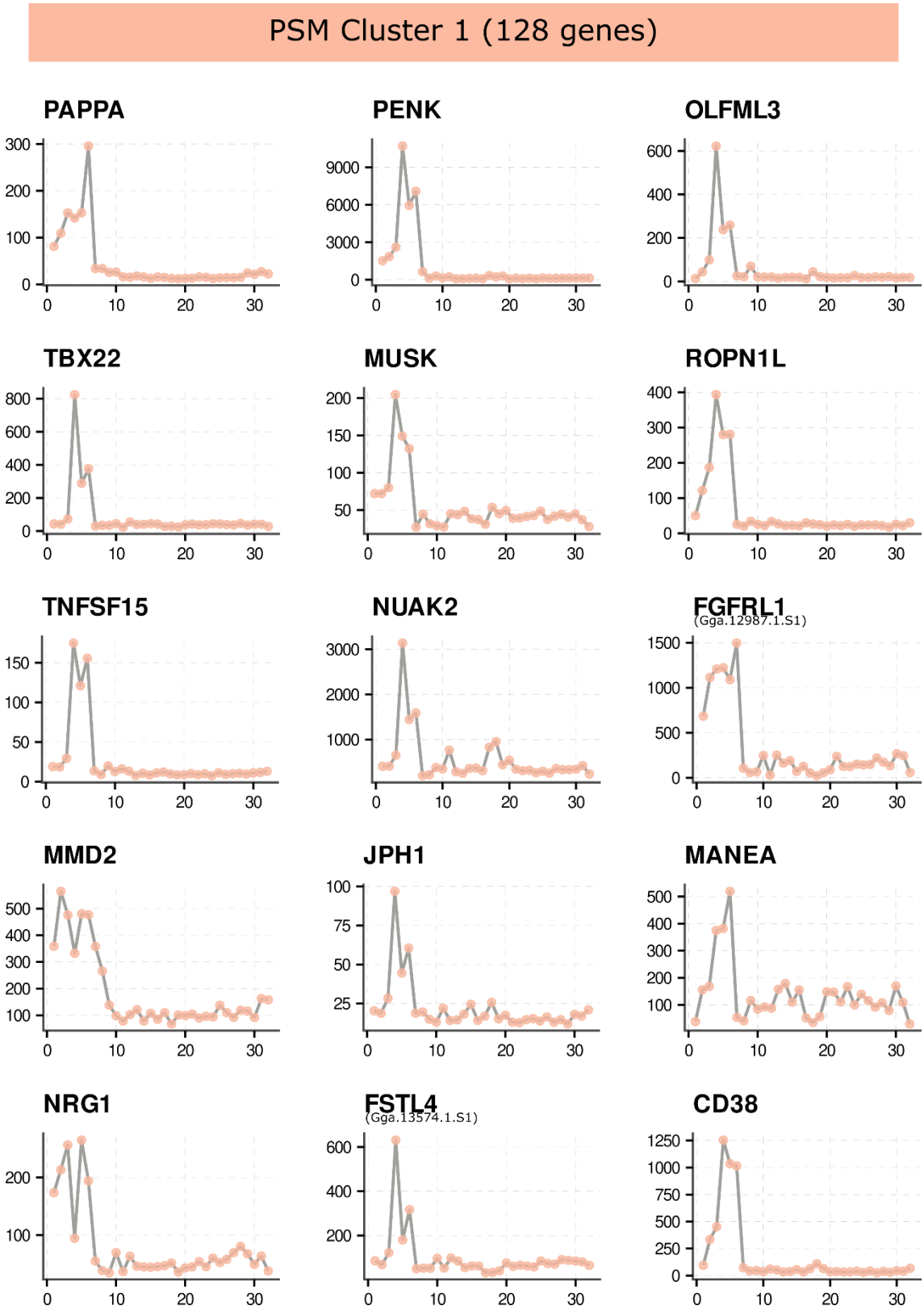
72	Gga.4131.1.S1_at	ENSGALG00000032039	**HOXD13**	BP, MF, CC
73	Gga.4154.2.S1_a_at	ENSGALG00000040620	**LSAMP**	BP, CC
74	Gga.4722.1.S1_s_at	ENSGALG00000002447	**CTNNA1**	
75	Gga.4890.2.S1_a_at	ENSGALG00000029244	**SMAP2**	BP, MF
76	Gga.4974.1.S1_at	ENSGALG00000015624	**VCAN**	BP, MF, CC
77	Gga.6252.1.S1_at	ENSGALG00000041808	**ANO1**	
78	GgaAffx.4752.1.S1_s_at			
79	Gga.645.1.S1_s_at	ENSGALG00000009356	**SKIL**	
80	Gga.7223.1.A1_s_at	ENSGALG00000000848	**AP1G1**	
81	Gga.738.1.S1_at	ENSGALG00000010974	**GSC**	BP, MF, CC
82	Gga.7683.1.S1_at	ENSGALG00000009302	**EMX2**	
83	Gga.7758.1.S1_at	ENSGALG00000010470	**TCERG1L**	
84	Gga.8121.1.S1_s_at	ENSGALG00000002336	**EP400**	
85	Gga.8371.1.S1_s_at	ENSGALG00000035533	**ADIPOR2**	
86	Gga.8407.1.S1_s_at	ENSGALG00000008827	**FERMT1**	
87	Gga.8999.1.S1_at	ENSGALG00000006840	**AKAP13**	
88	GgaAffx.22069.1.S1_s_at			
89	Gga.9080.1.S1_at	ENSGALG00000012913	**PKP2**	
90	Gga.958.1.S1_at	ENSGALG00000039629	**HOXD11**	BP, MF, CC
91	Gga.9659.1.S1_s_at	ENSGALG00000029512	**SPTBN1**	
92	GgaAffx.22491.4.S1_s_at			
93	GgaAffx.22491.2.S1_s_at			
94	Gga.969.1.S1_at	ENSGALG00000002911	**MOXD1**	BP, MF, CC
95	GgaAffx.10417.7.S1_s_at	ENSGALG00000016289	**DST**	
96	GgaAffx.10904.1.S1_at	ENSGALG00000017026	**VPS36**	
97	GgaAffx.12237.1.S1_s_at	ENSGALG00000005425	**OGT**	
98	GgaAffx.12773.1.S1_s_at	ENSGALG00000042458	**ACTN1**	BP, MF, CC
99	GgaAffx.1652.1.S1_s_at	ENSGALG00000002469	**CDC73**	BP, MF, CC
100	GgaAffx.1660.2.S1_s_at	ENSGALG00000002499	**ZMYM4**	
101	GgaAffx.20432.1.S1_at	ENSGALG00000002242	**GALNT9**	
102	GgaAffx.20550.1.S1_s_at	ENSGALG00000041977	**RNF128**	
103	GgaAffx.20794.1.S1_s_at	--	**LOC415456**	
104	GgaAffx.2100.1.S1_at	--	**USP42**	
105	GgaAffx.21048.1.S1_s_at	ENSGALG00000037618	**ASAP1**	
106	GgaAffx.21250.1.S1_s_at	ENSGALG00000029834	**C9ORF58**	
107	GgaAffx.21469.1.S1_s_at	ENSGALG00000004088	**FNBP1**	
108	GgaAffx.21483.1.S1_at	ENSGALG00000016886	**FARP1**	BP, MF, CC
109	GgaAffx.21783.1.S1_s_at	ENSGALG00000030530	**HAND2**	BP, MF, CC
110	GgaAffx.21899.1.S1_at	ENSGALG00000052354	**SON**	
111	GgaAffx.21899.3.S1_s_at			
112	GgaAffx.21976.1.S1_s_at	ENSGALG00000006568	**TM9SF4**	
113	GgaAffx.22296.1.S1_s_at	ENSGALG00000033469	**KDM1A**	

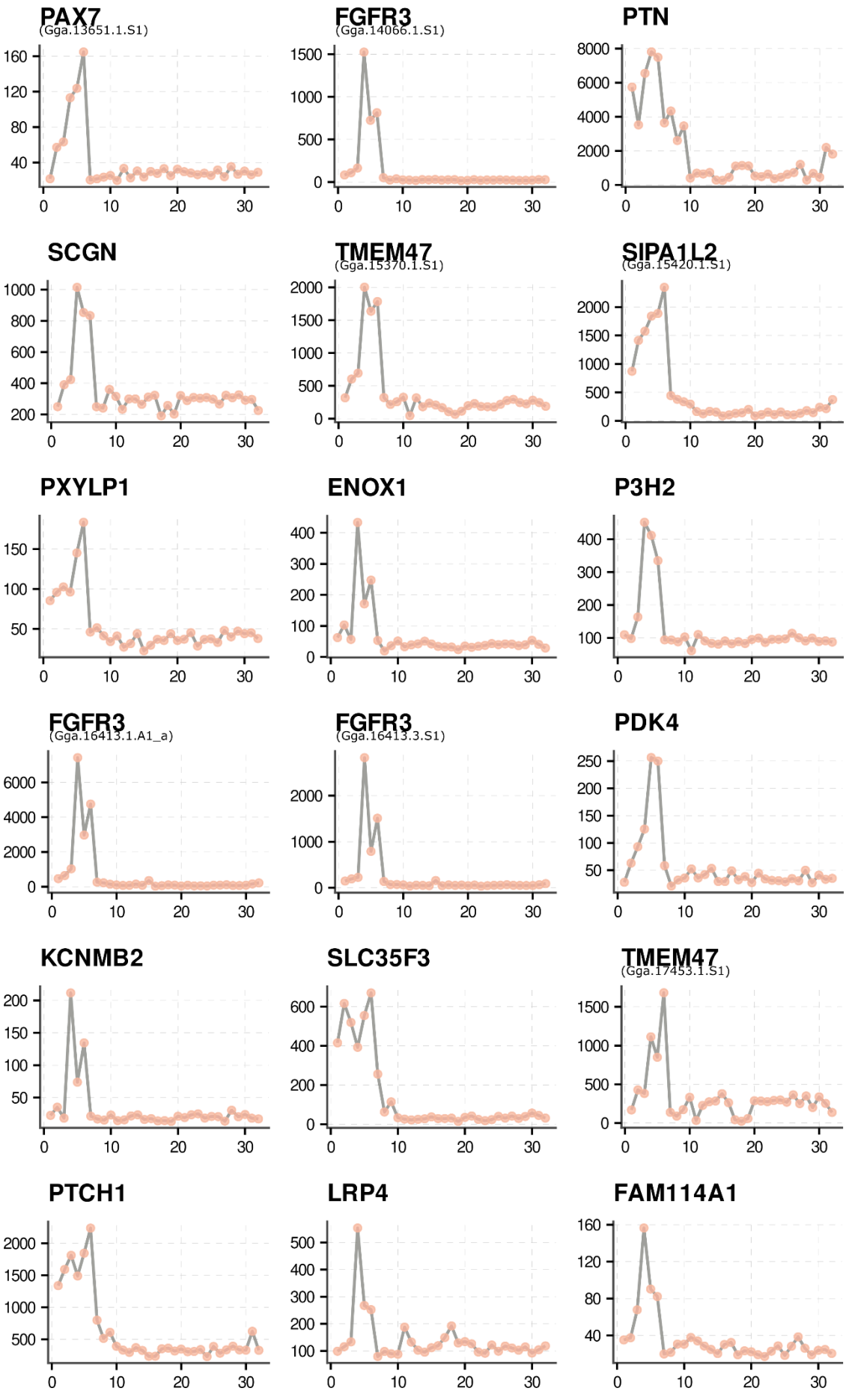
114	GgaAffx.22675.1.S1_at	ENSGALG00000008725	**KIF16B**	
115	GgaAffx.22899.1.S1_s_at	ENSGALG00000001030	**SCAMP4**	
116	GgaAffx.22924.2.S1_s_at	ENSGALG00000041034	**MIA3**	
117	GgaAffx.2298.1.S1_at	ENSGALG00000017414	**JUP**	
118	GgaAffx.2326.1.S1_at	ENSGALG00000003717	**MYO9B**	
119	GgaAffx.23939.1.S1_s_at	ENSGALG00000012543	**GPD2**	
120	GgaAffx.23950.1.S1_s_at	ENSGALG00000012627	**CDC14B**	
121	GgaAffx.24078.5.S1_s_at	ENSGALG00000031495	**TRIO**	
122	GgaAffx.24165.1.S1_at	ENSGALG00000013208	**CENPE**	
123	GgaAffx.25074.1.S1_s_at	ENSGALG00000016781	**MAP4K4**	
124	GgaAffx.25349.2.S1_s_at	ENSGALG00000039068	**RSF1**	
125	GgaAffx.25358.2.S1_s_at	ENSGALG00000001995	**LIMS2**	
126	GgaAffx.25567.1.S1_at	--	**NEFL**	
127	GgaAffx.25594.1.S1_at	ENSGALG00000002719	**CSNK1D**	
128	GgaAffx.25606.1.S1_at	ENSGALG00000037625	**FBN3**	
129	GgaAffx.25873.1.S1_at	ENSGALG00000031388	**KLHL5**	
130	GgaAffx.25899.1.S1_s_at	ENSGALG00000003596	**TRRAP**	
131	GgaAffx.26.1.S1_at	ENSGALG00000034003	**ARID4B**	
132	GgaAffx.26247.1.S1_at	ENSGALG00000004573	**LRRC8A**	
133	GgaAffx.26339.1.S1_s_at	ENSGALG00000004852	**DNM1**	
134	GgaAffx.26544.1.S1_at	ENSGALG00000005618	**CTR9**	
135	GgaAffx.26551.2.S1_s_at	ENSGALG00000005637	**GBF1**	
136	GgaAffx.2835.1.S1_at	ENSGALG00000004587	**RC3H1**	
137	GgaAffx.3133.1.S1_s_at	ENSGALG00000005028	**TOP3A**	
138	GgaAffx.3145.1.S1_s_at	ENSGALG00000005035	**KAT6B**	
139	GgaAffx.3294.1.S1_at	ENSGALG00000005265	**SYT9**	
140	GgaAffx.3545.2.S1_s_at	ENSGALG00000005646	**TSC2**	
141	GgaAffx.3564.1.S1_s_at	ENSGALG00000005683	**ARHGAP29**	
142	GgaAffx.4438.1.S1_at	ENSGALG00000007097	**RPS6KA6**	
143	GgaAffx.4438.2.S1_s_at			
144	GgaAffx.4681.1.S1_at	ENSGALG00000031303	**FAM234A**	
145	GgaAffx.4808.1.S1_s_at	ENSGALG00000007721	**ARHGAP21**	
146	GgaAffx.5198.1.S1_s_at	ENSGALG00000008238	**MED13L**	
147	GgaAffx.5721.1.S1_at	ENSGALG00000009056	**LCLAT1**	BP, MF, CC
148	GgaAffx.6023.1.S1_at	ENSGALG00000009595	**TWF1**	
149	GgaAffx.6136.1.S1_at	ENSGALG00000041494	**GOLGB1**	
150	GgaAffx.6223.1.S1_at	ENSGALG00000009907	**G2E3**	BP, MF, CC
151	GgaAffx.654.1.S1_at	ENSGALG00000001019	**BTBD2**	
152	GgaAffx.6983.1.S1_at	ENSGALG00000039349	**DISC1**	
153	GgaAffx.7190.1.S1_s_at	ENSGALG00000011309	**PLXNC1**	
154	GgaAffx.7307.1.S1_at	ENSGALG00000011532	**CYP27C1**	
155	GgaAffx.736.3.S1_s_at	ENSGALG00000001110	**TYW1**	

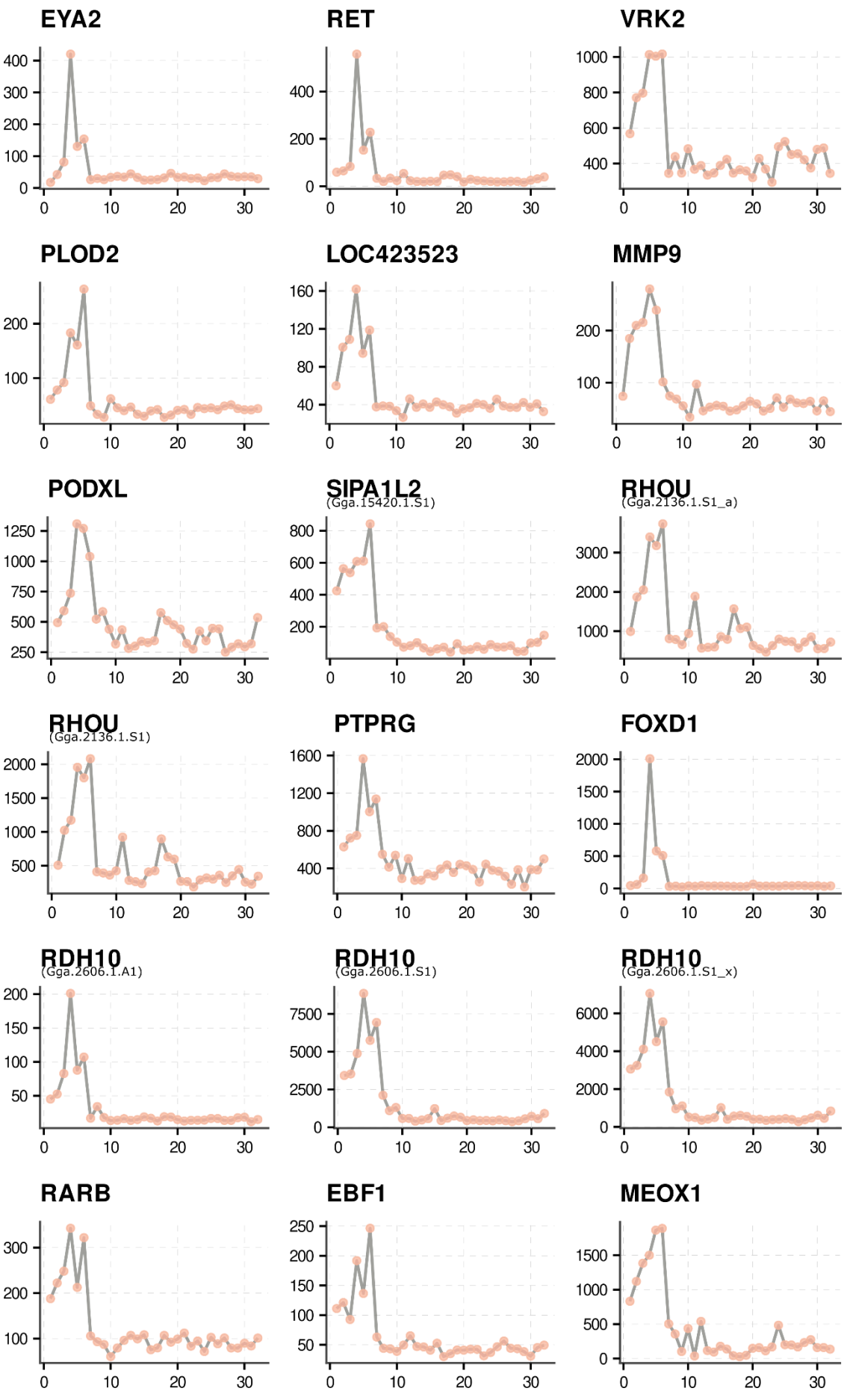
156	GgaAffx.7823.1.S1_s_at	ENSGALG00000031117	**STK17A**	
157	GgaAffx.816.3.S1_s_at	ENSGALG00000001229	**SKI**	BP, MF, CC
158	GgaAffx.8204.2.S1_s_at	ENSGALG00000012966	**ZBTB2**	
159	GgaAffx.8525.8.S1_s_at	ENSGALG00000038924	**IDE**	
160	GgaAffx.8673.1.S1_at	ENSGALG00000013616	**OPRM1**	
161	GgaAffx.8701.1.S1_s_at	ENSGALG00000013659	**NOX3**	
162	GgaAffx.8702.1.S1_at	ENSGALG00000013660	**ZNF516**	
163	GgaAffx.9141.1.S1_at	ENSGALG00000036085	**LOC776273**	
164	GgaAffx.9169.1.S1_at	ENSGALG00000045611	**GLMP**	
165	GgaAffx.9263.1.S1_s_at	ENSGALG00000039160	**SUPT5H**	BP, MF, CC
166	GgaAffx.9480.1.A1_at	ENSGALG00000014930	**ENC1**	
167	GgaAffx.9495.1.S1_at	ENSGALG00000014944	**GCNT4**	
168	GgaAffx.9512.2.S1_s_at	ENSGALG00000014974	**MIB1**	
169	GgaAffx.9521.1.S1_at	ENSGALG00000014995	**PGM5**	
170	GgaAffx.9522.1.S1_at			
171	GgaAffx.9840.2.S1_s_at	ENSGALG00000015464	**PTGFRN**	
172	GgaAffx.990.1.S1_at	ENSGALG00000035187	**FHL3**	
173	GgaAffx.9910.1.S1_s_at	ENSGALG00000015551	**GRIN3A**	

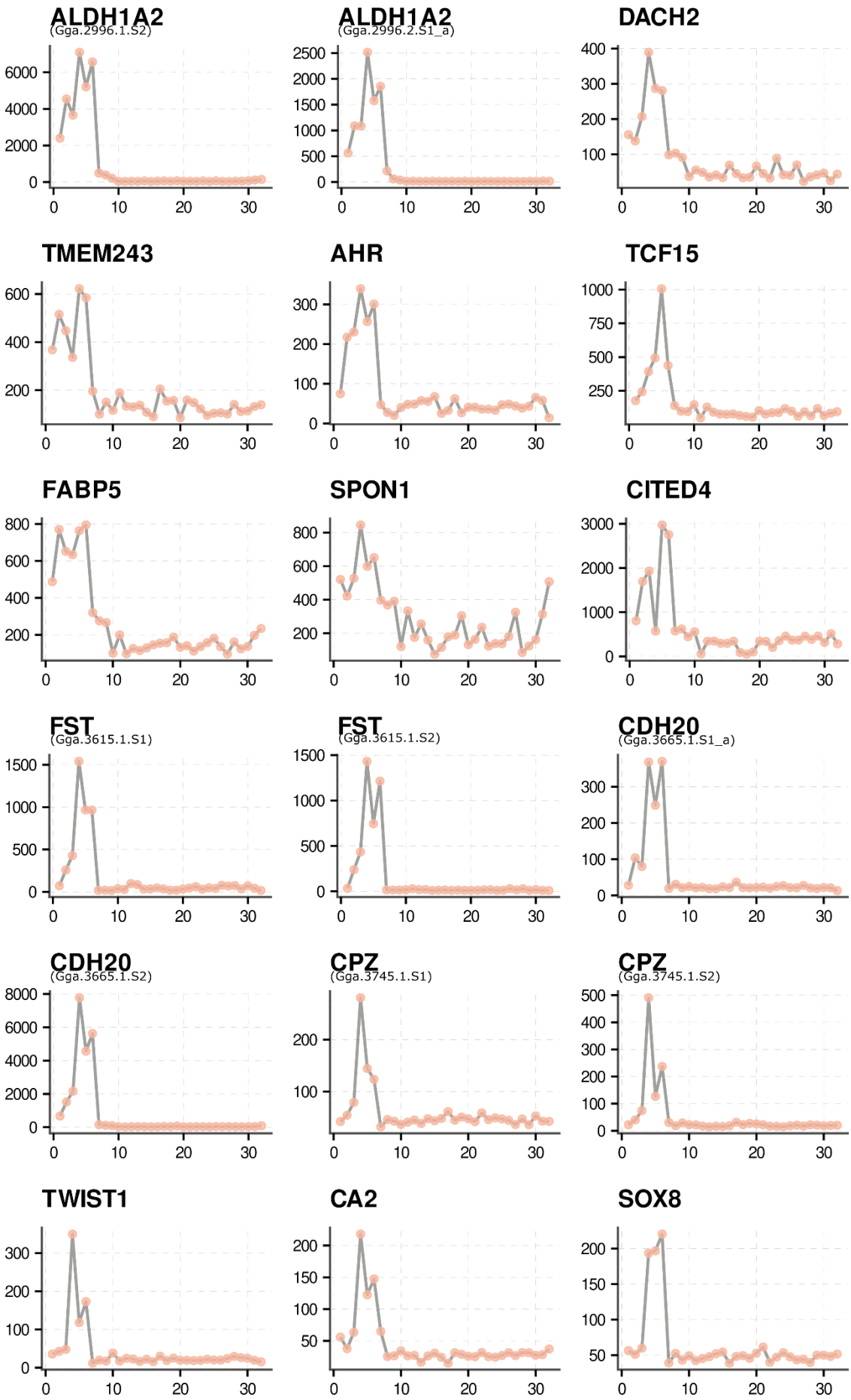
Annex 7 | Pseudo-temporal trajectories of the ClockOME genes

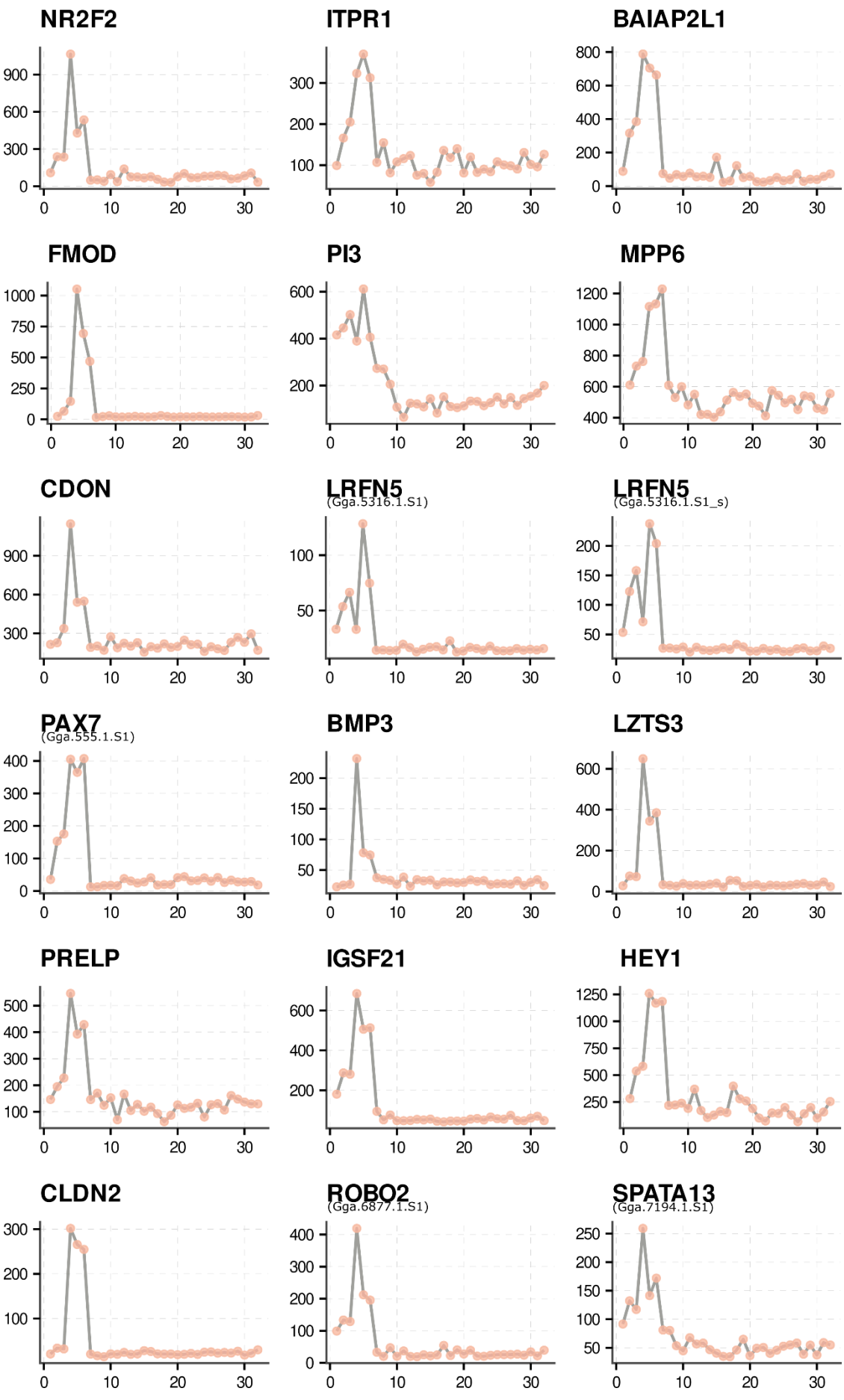
Annex 7.1 - Pseudo-temporal trajectories of the genes in the PSM K1

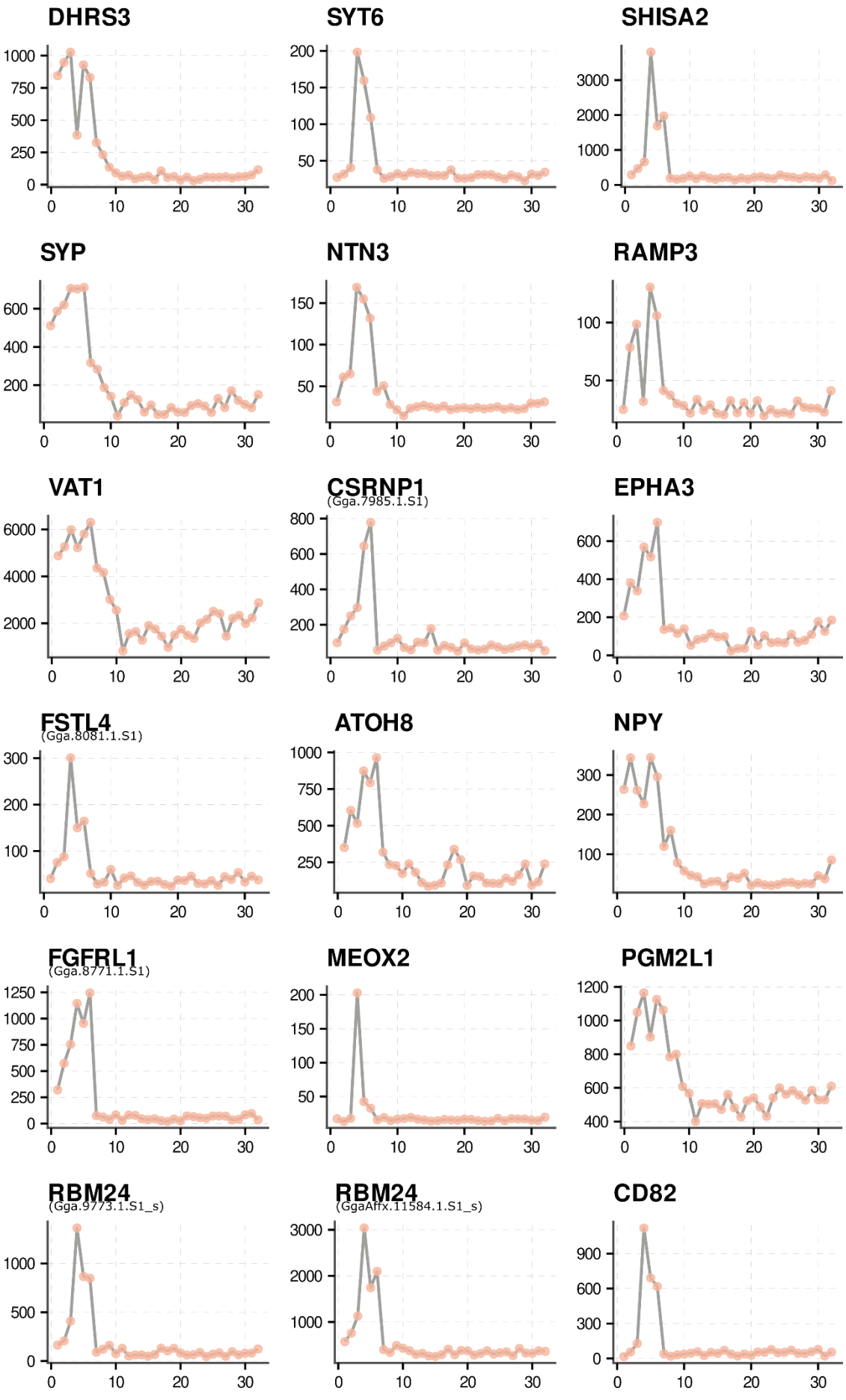


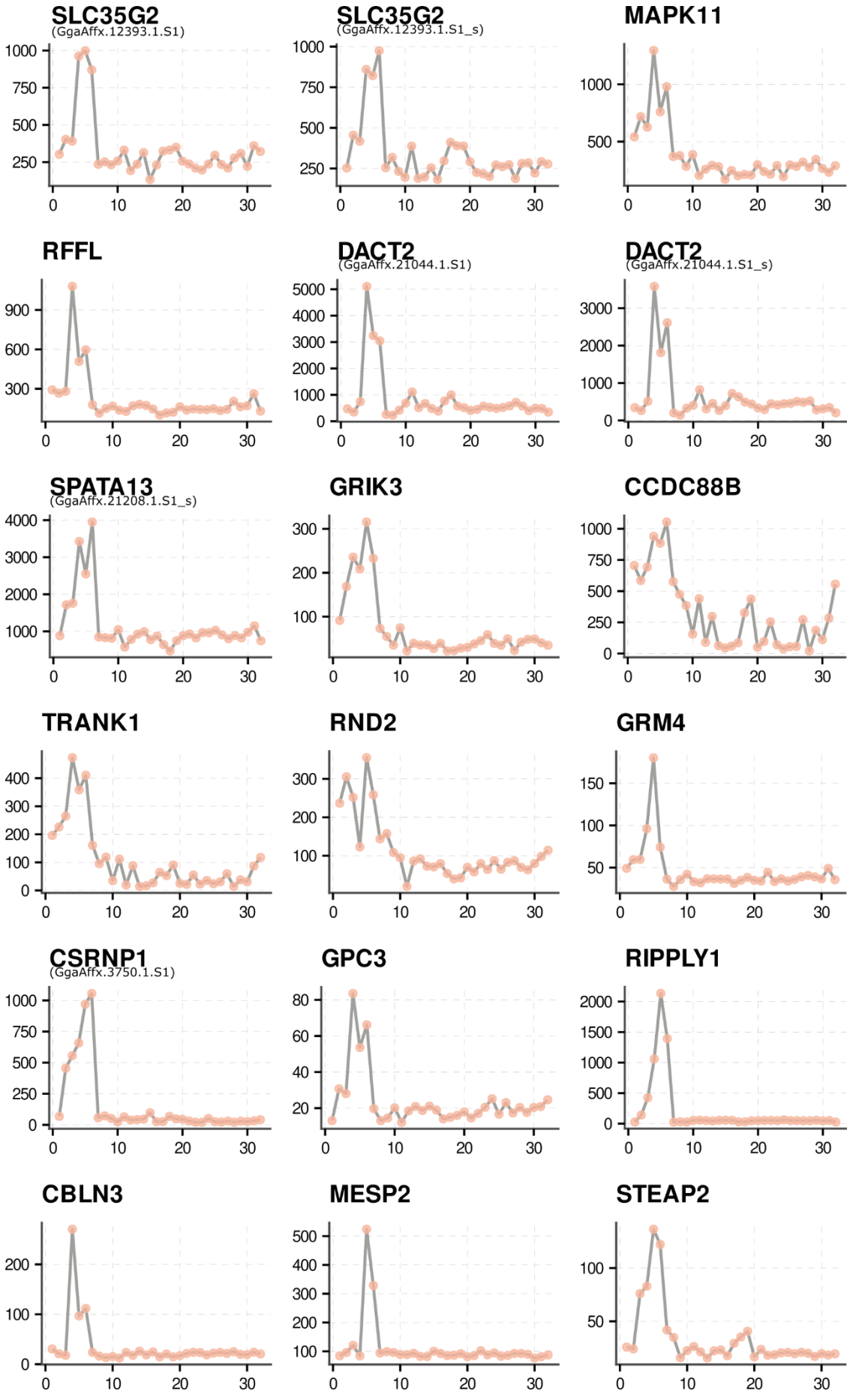

PAX7
(Gga.13651.1.S1)
FGFR3
(Gga.14066.1.S1)
PTN
SCGN
TMEM47
(Gga.15370.1.S1)
SIPA1L2
(Gga.15420.1.S1)
PXYLP1
ENOX1
P3H2
FGFR3
(Gga.16413.1.A1_a)
FGFR3
(Gga.16413.3.S1)
PDK4
KCNMB2
SLC35F3
TMEM47
(Gga.17453.1.S1)
PTCH1
LRP4
FAM114A1

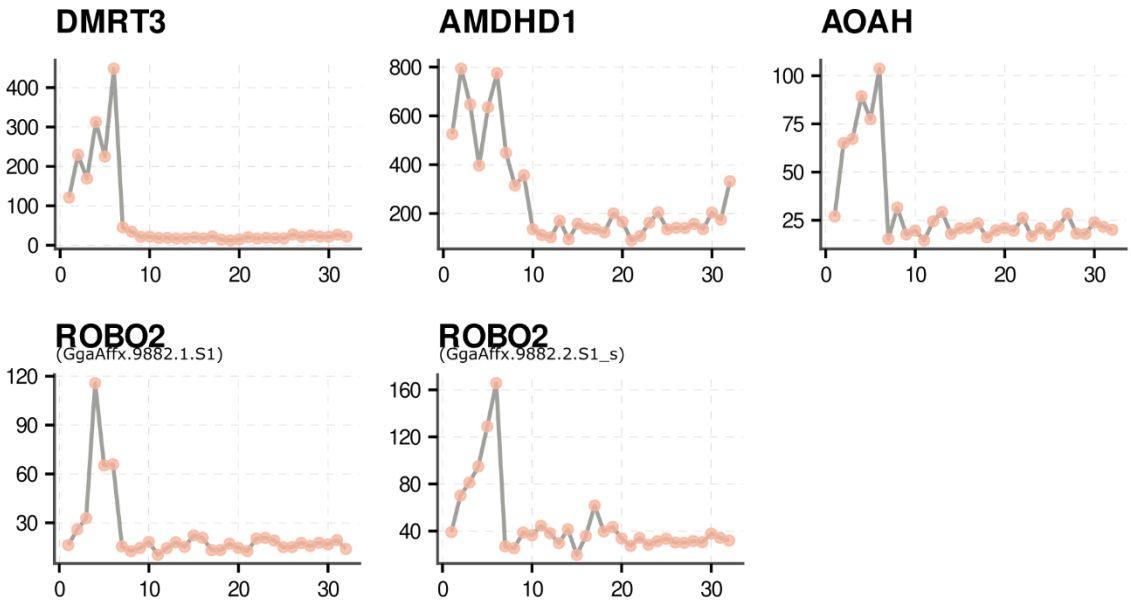

EYA2
RET
VRK2
PLOD2
LOC423523
MMP9
PODXL
SIPA1L2
(Gga.15420.1.S1)
RHOU
(Gga.2136.1.S1_a)
RHOU
(Gga.2136.1.S1)
PTPRG
FOXD1
RDH10
(Gga.2606.1.A1)
RDH10
(Gga.2606.1.S1)
RDH10
(Gga.2606.1.S1_x)
RARB
EBF1
MEOX1


ALDH1A2
(Gga.2996.1.S2)
ALDH1A2
(Gga.2996.2.S1_a)
DACH2
TMEM243
AHR
TCF15
FABP5
SPON1
CITED4
FST
(Gga.3615.1.S1)
FST
(Gga.3615.1.S2)
CDH20
(Gga.3665.1.S1_a)
CDH20
(Gga.3665.1.S2)
CPZ
(Gga.3745.1.S1)
CPZ
(Gga.3745.1.S2)
TWIST1
CA2
SOX8

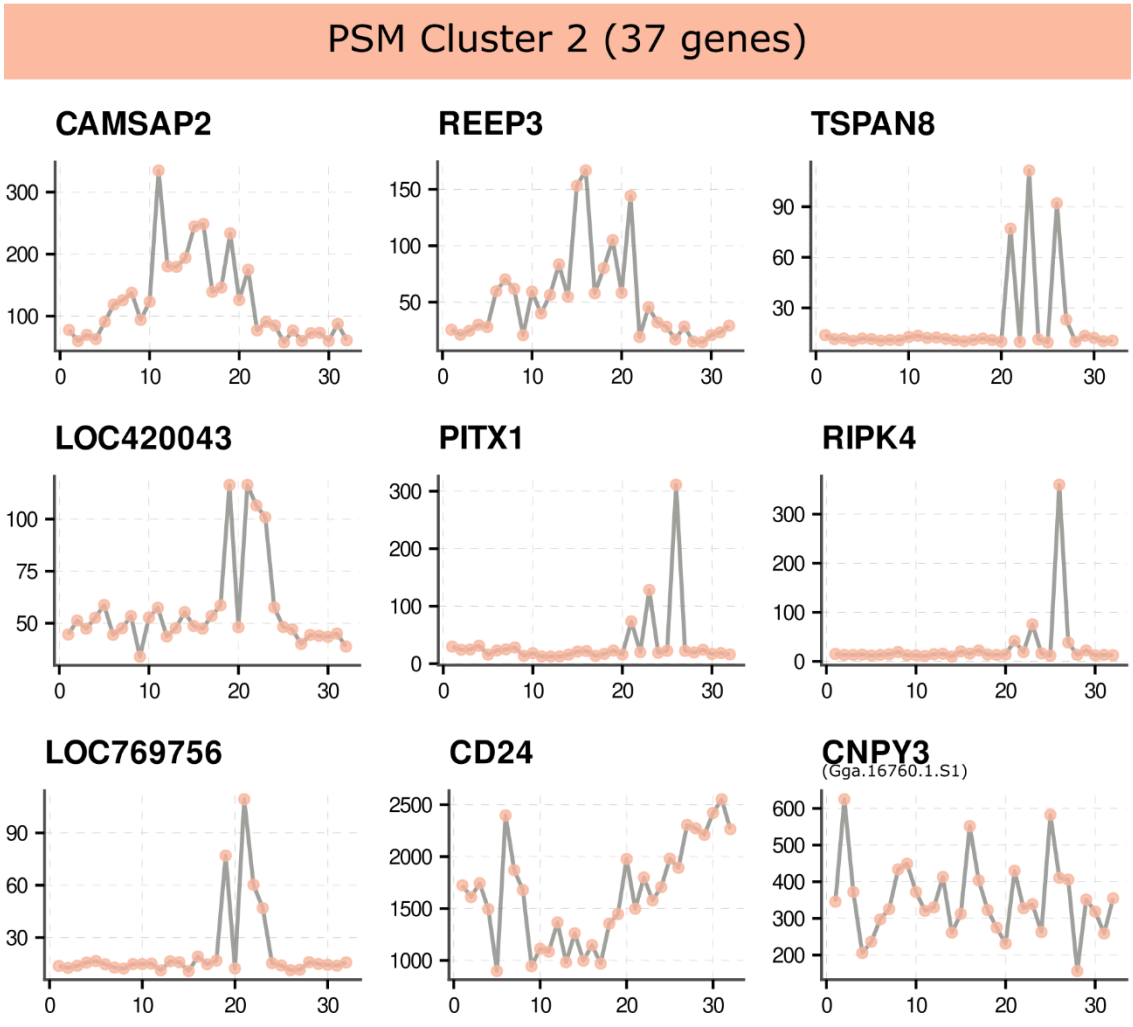

NR2F2
ITPR1
BAIAP2L1
FMOD
PI3
MPP6
CDON
LRFN5
(Gga.5316.1.S1)
LRFN5
(Gga.5316.1.S1_s)
PAX7
(Gga.555.1.S1)
BMP3
LZTS3
PRELP
IGSF21
HEY1
CLDN2
ROBO2
(Gga.6877.1.S1)
SPATA13
(Gga.7194.1.S1)

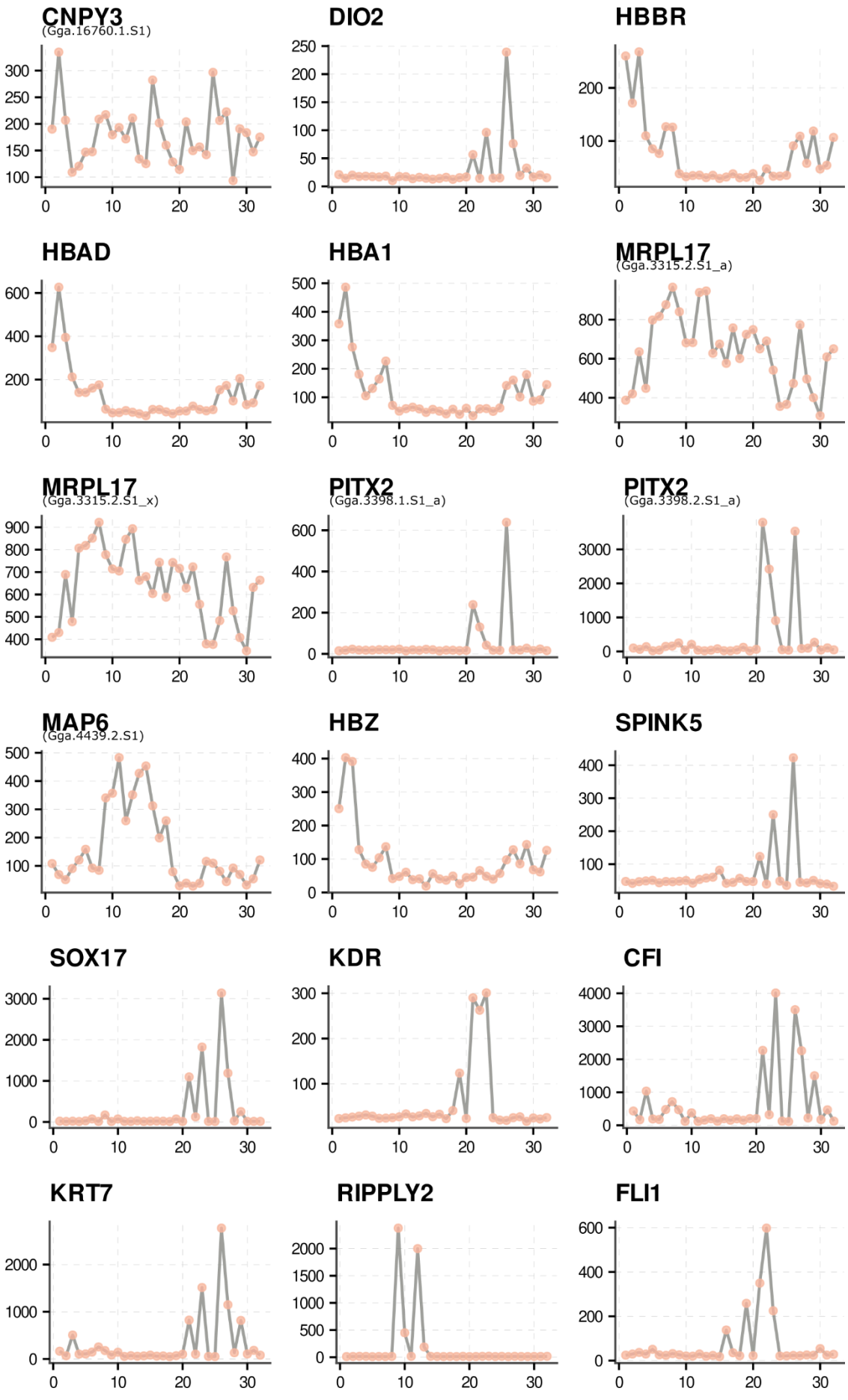

DHRS3
SYT6
SHISA2
SYP
NTN3
RAMP3
VAT1
CSRNP1
(Gga.7985.1.S1)
EPHA3
FSTL4
(Gga.8081.1.S1)
ATOH8
NPY
FGFRL1
(Gga.8771.1.S1)
MEOX2
PGM2L1
RBM24
(Gga.9773.1.S1_s)
RBM24
(GgaAffx.11584.1.S1_s)
CD82

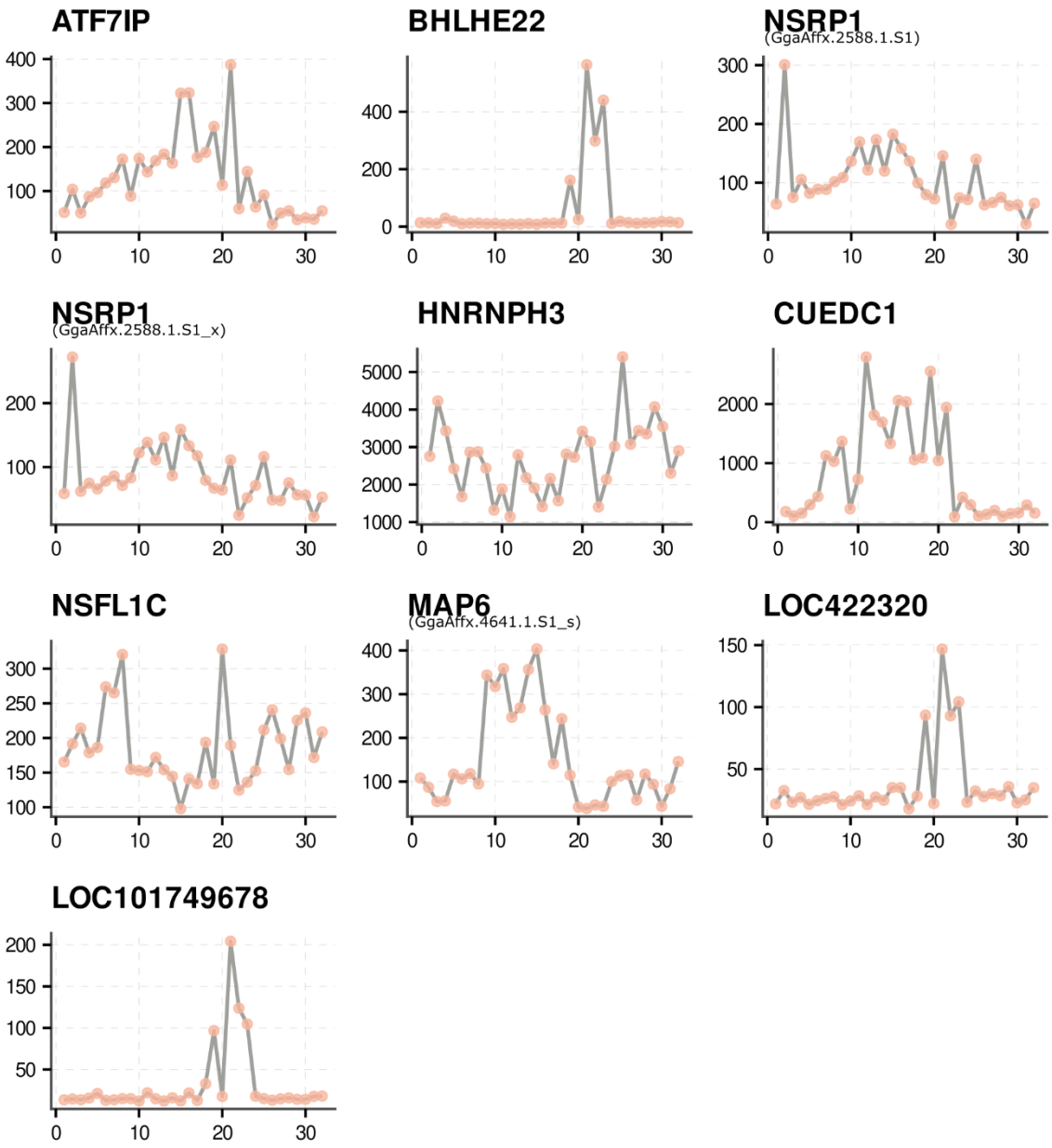

SLC35G2
(GgaAffx.12393.1.S1)
SLC35G2
(GgaAffx.12393.1.S1_s)
MAPK11
RFFL
DACT2
(GgaAffx.21044.1.S1)
DACT2
(GgaAffx.21044.1.S1_s)
SPATA13
(GgaAffx.21208.1.S1_s)
GRIK3
CCDC88B
TRANK1
RND2
GRM4
CSRNP1
(GgaAffx.3750.1.S1)
GPC3
RIPPLY1
CBLN3
MESP2
STEAP2



Annex 7.2 - Pseudo-temporal trajectories of the genes in the PSM K2

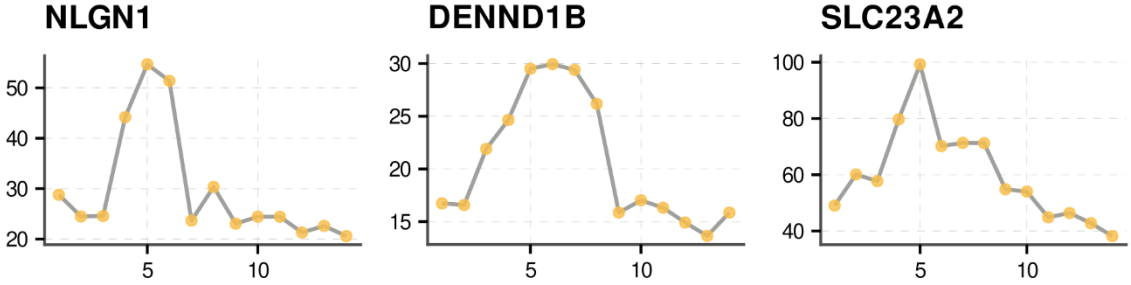


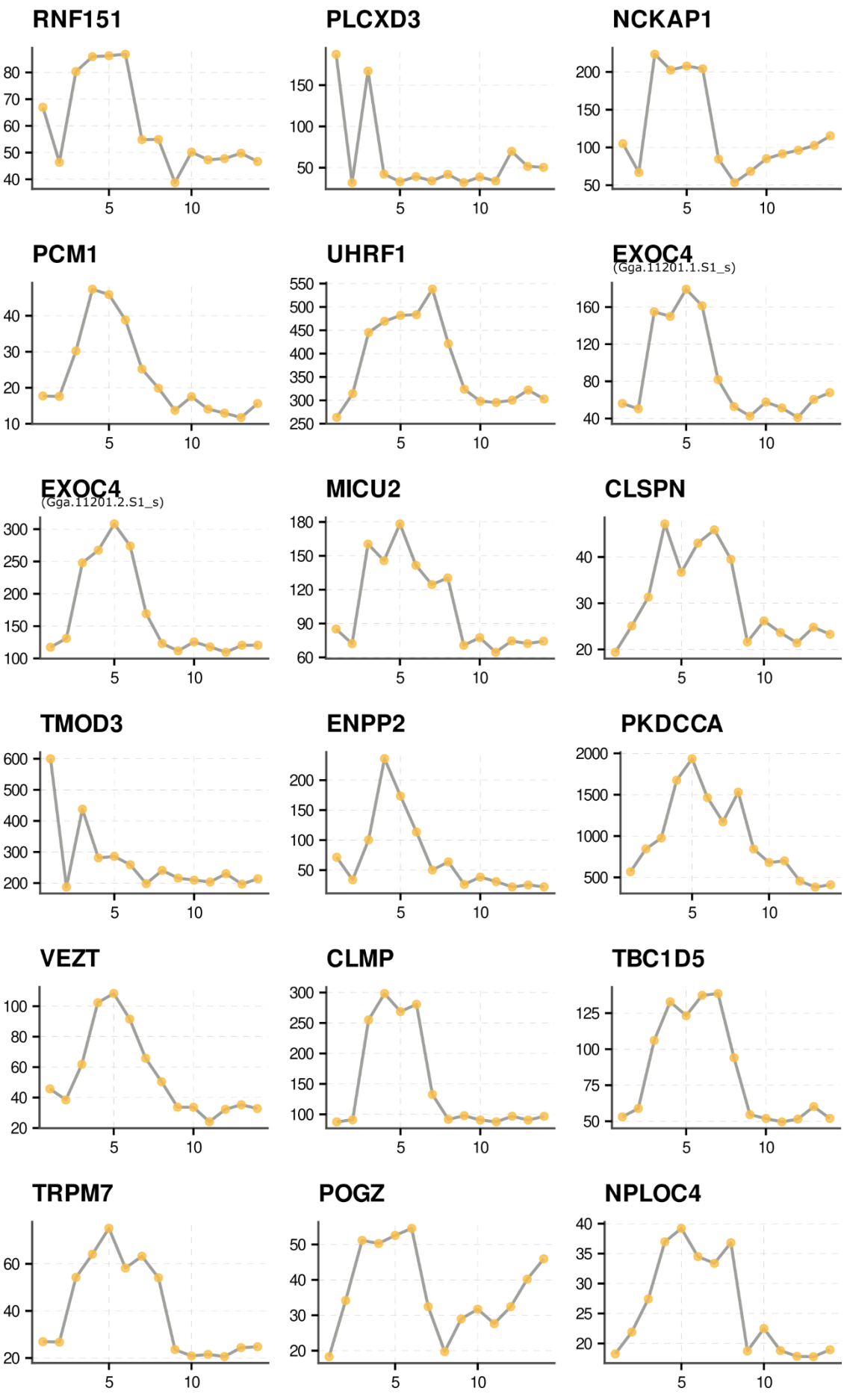

CNPY3
(Gga.16760.1.S1)
DIO2
HBBR
HBAD
HBA1
MRPL17
(Gga.3315.2.S1_a)
MRPL17
(Gga.3315.2.S1_x)
PITX2
(Gga.3398.1.S1_a)
PITX2
(Gga.3398.2.S1_a)
MAP6
(Gga.4439.2.S1)
HBZ
SPINK5
SOX17
KDR
CFI
KRT7
RIPPLY2
FLI1

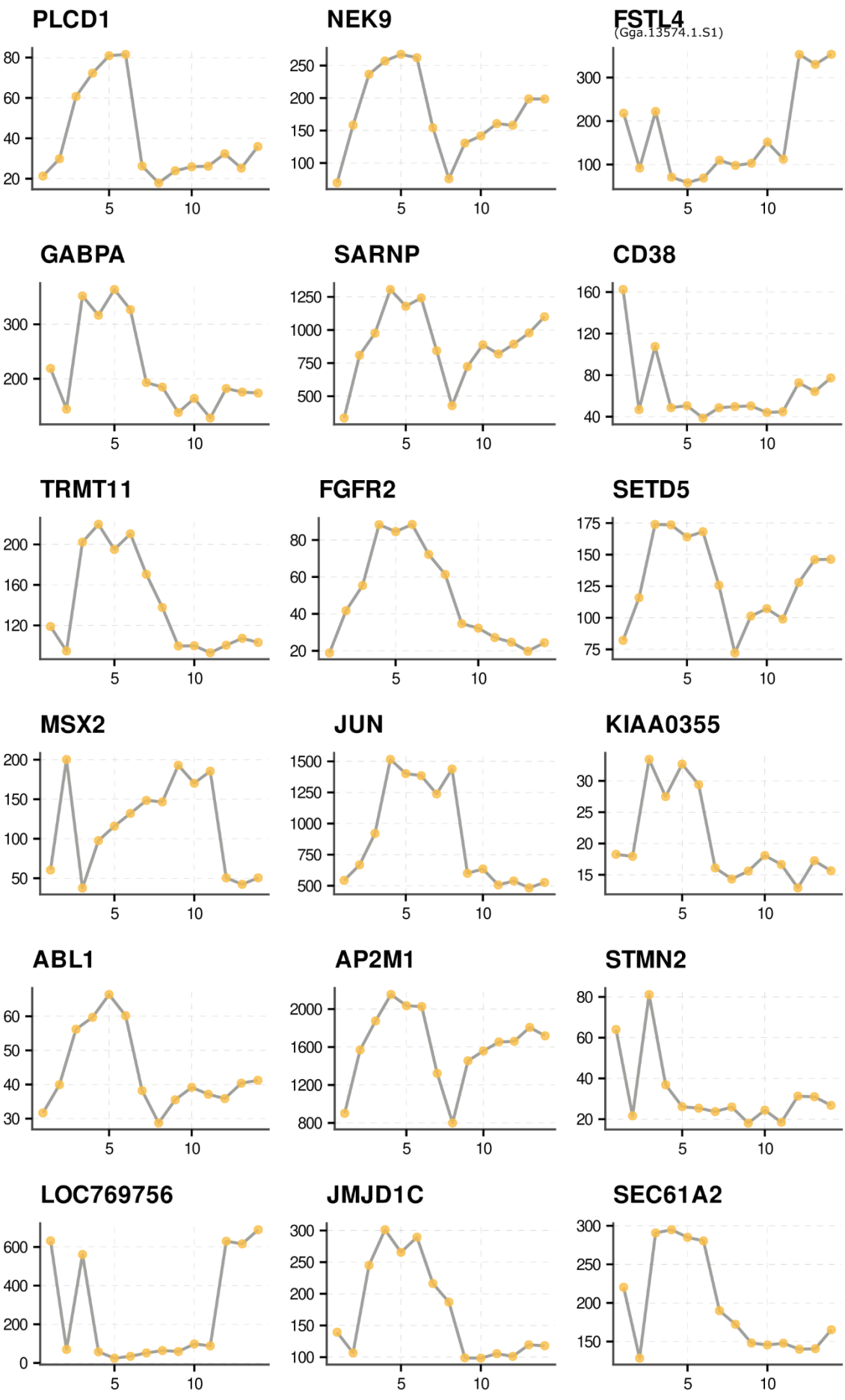


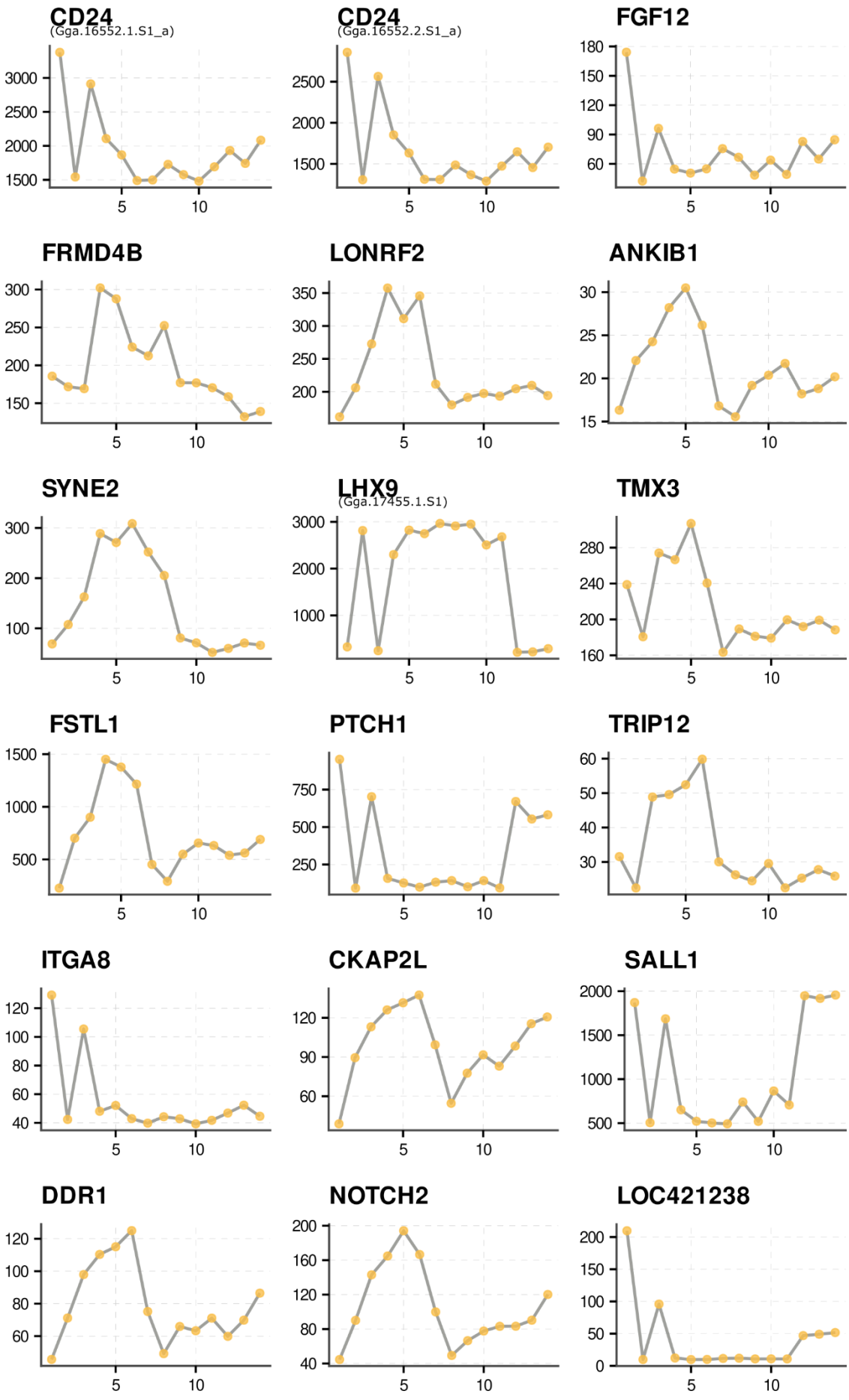
Annex 7.3 - Pseudo-temporal trajectories of the genes in the Limb K1

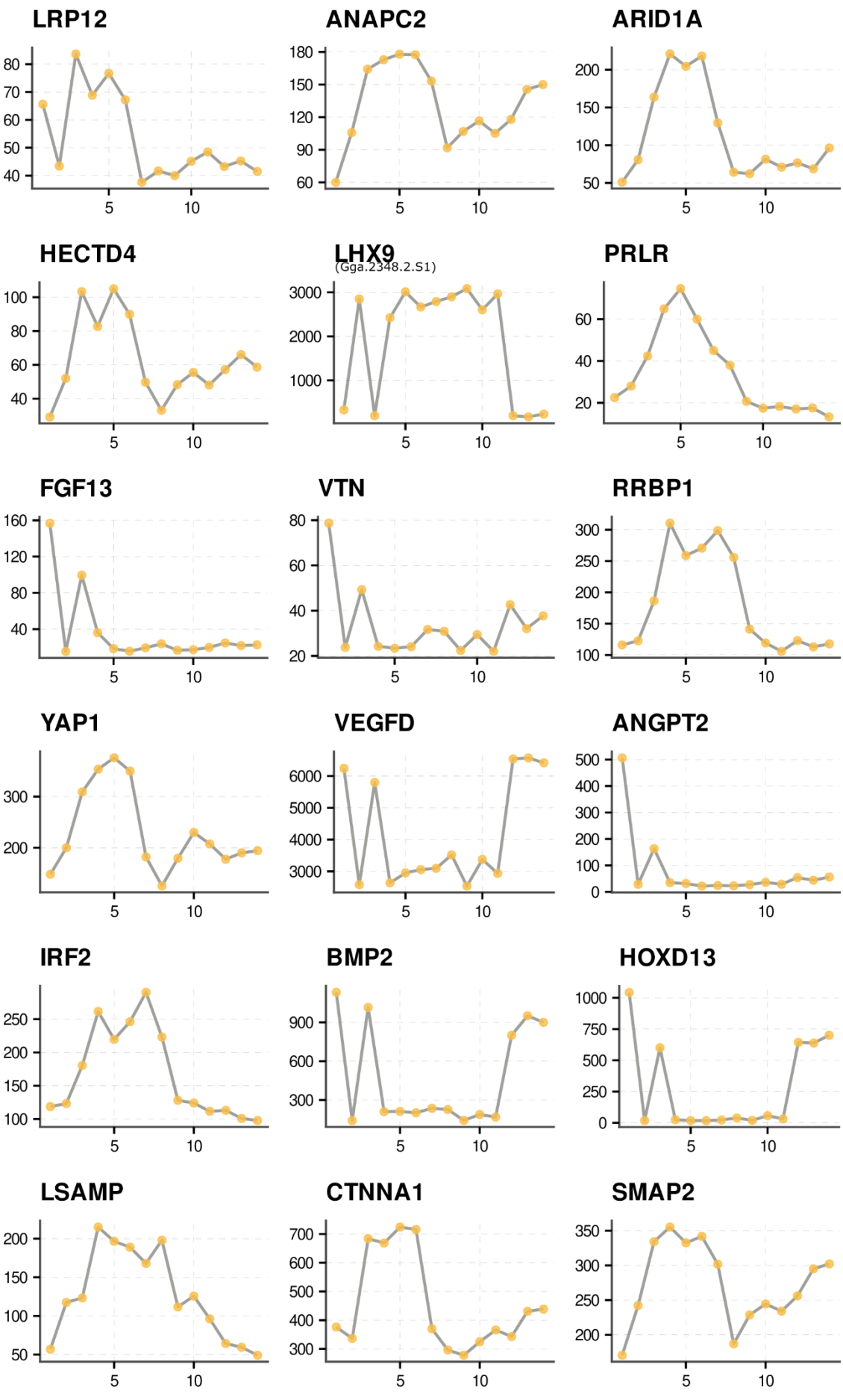
Limb Cluster (173 genes)

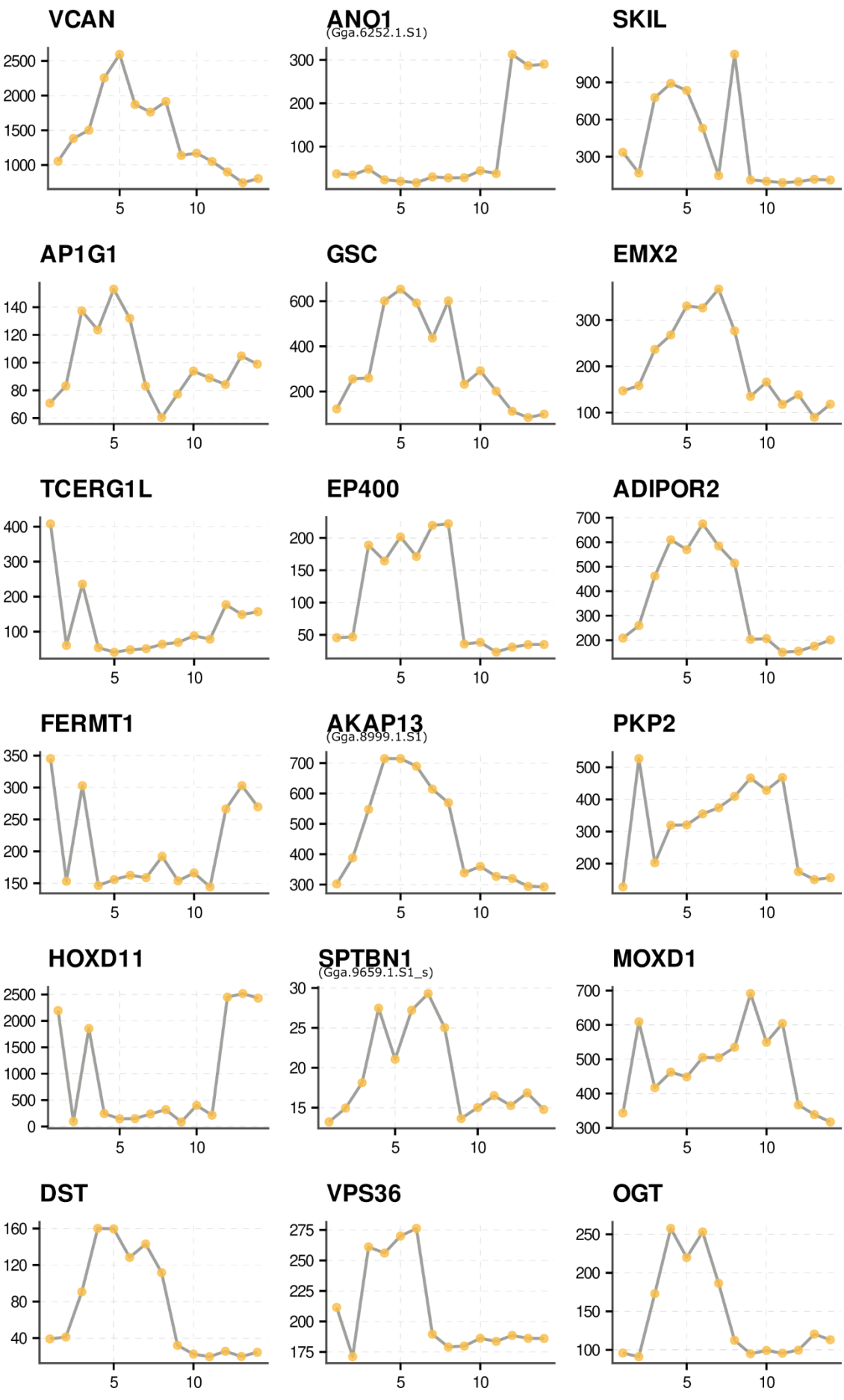


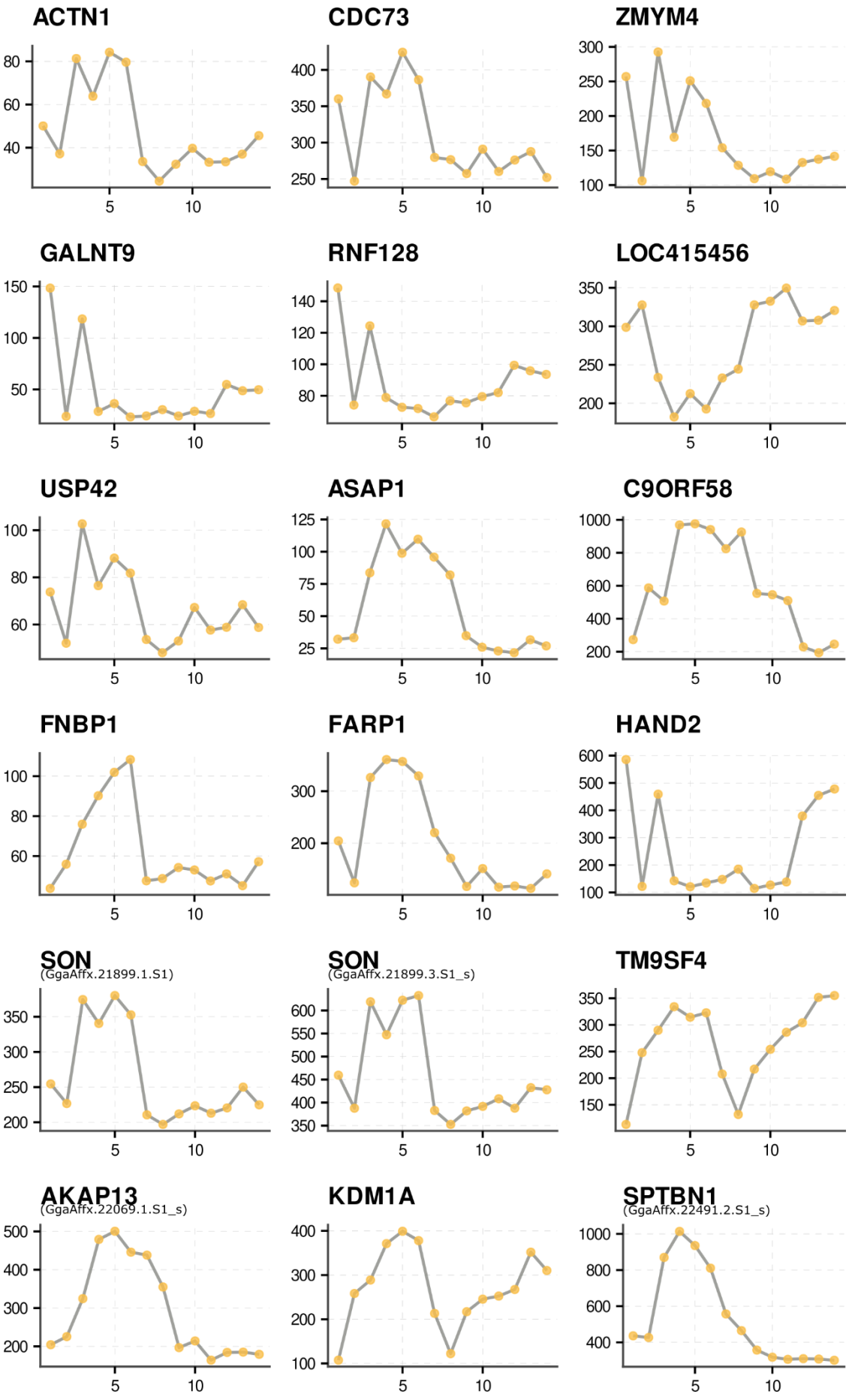

RNF151
PLCXD3
NCKAP1
PCM1
UHRF1
EXOC4
(Gga.11201.1.S1_s)
EXOC4
(Gga.11201.2.S1_s)
MICU2
CLSPN
TMOD3
ENPP2
PKDCCA
VEZT
CLMP
TBC1D5
TRPM7
POGZ
NPLOC4

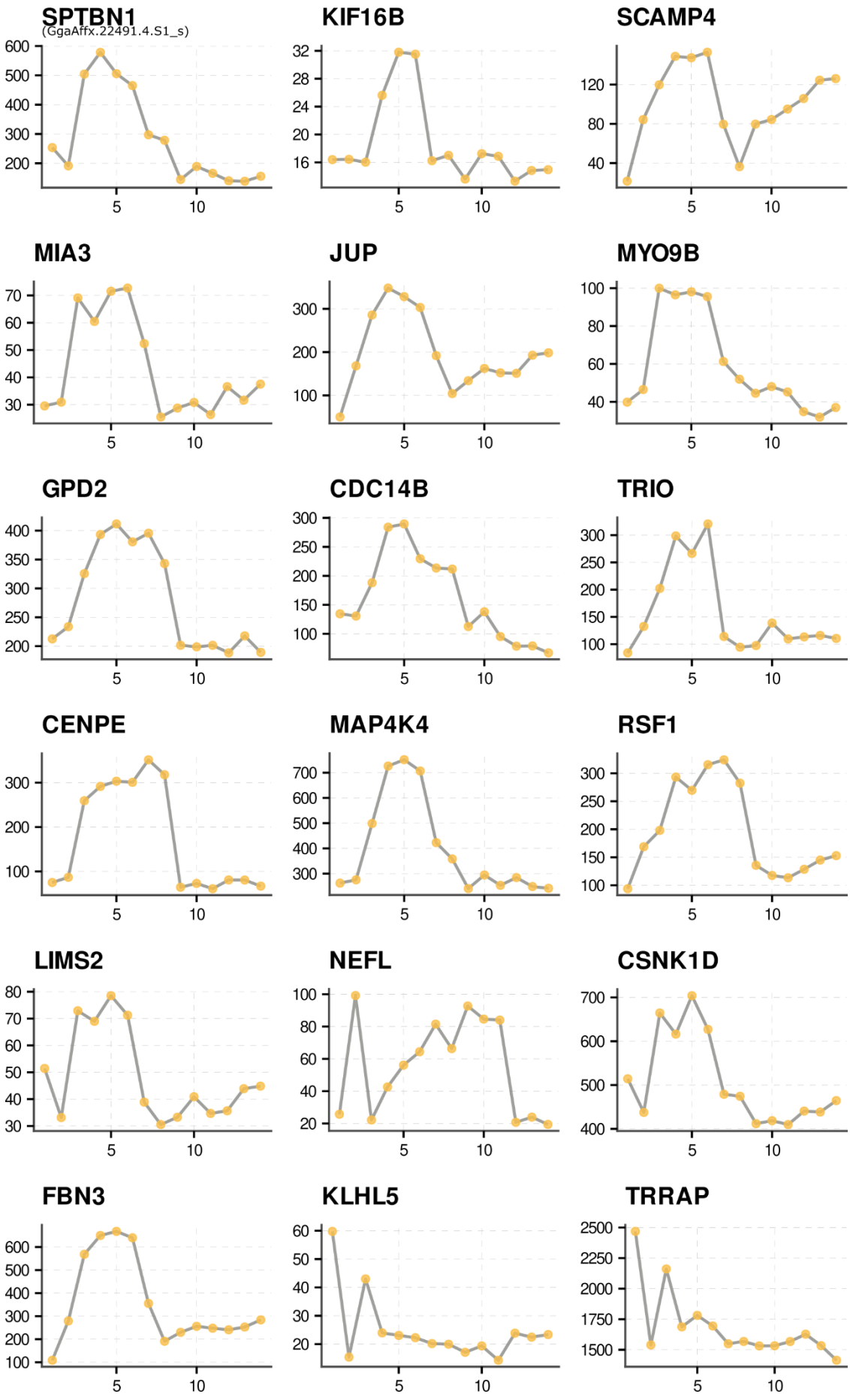

PLCD1
NEK9
FSTL4
(Gga.13574.1.S1)
GABPA
SARNP
CD38
TRMT11
FGFR2
SETD5
MSX2
JUN
KIAA0355
ABL1
AP2M1
STMN2
LOC769756
JMJD1C
SEC61A2

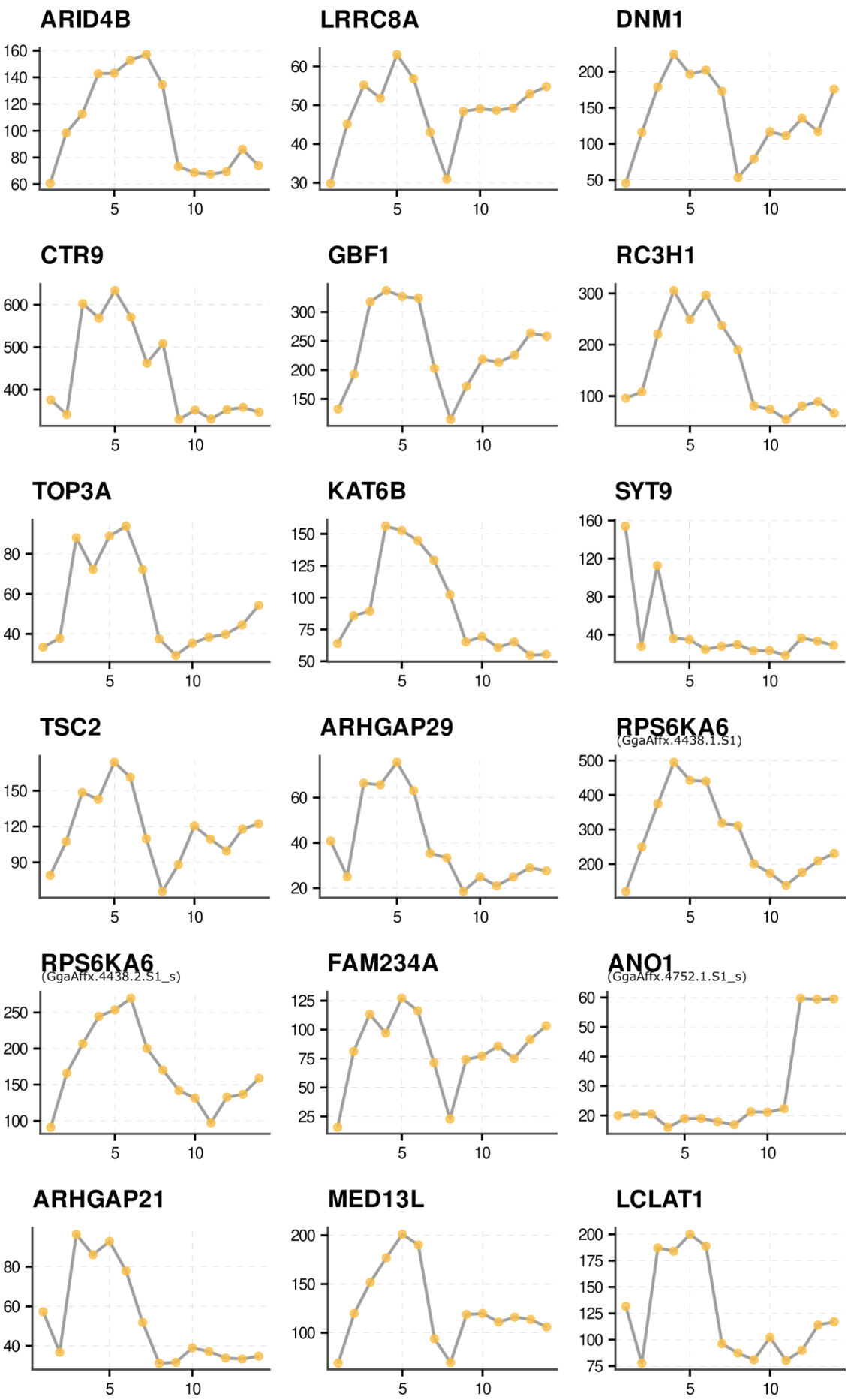

CD24
(Gga.16552.1.S1_a)
CD24
(Gga.16552.2.S1_a)
FGF12
FRMD4B
LONRF2
ANKIB1
SYNE2
LHX9
(Gga.17455.1.S1)
TMX3
FSTL1
PTCH1
TRIP12
ITGA8
CKAP2L
SALL1
DDR1
NOTCH2
LOC421238

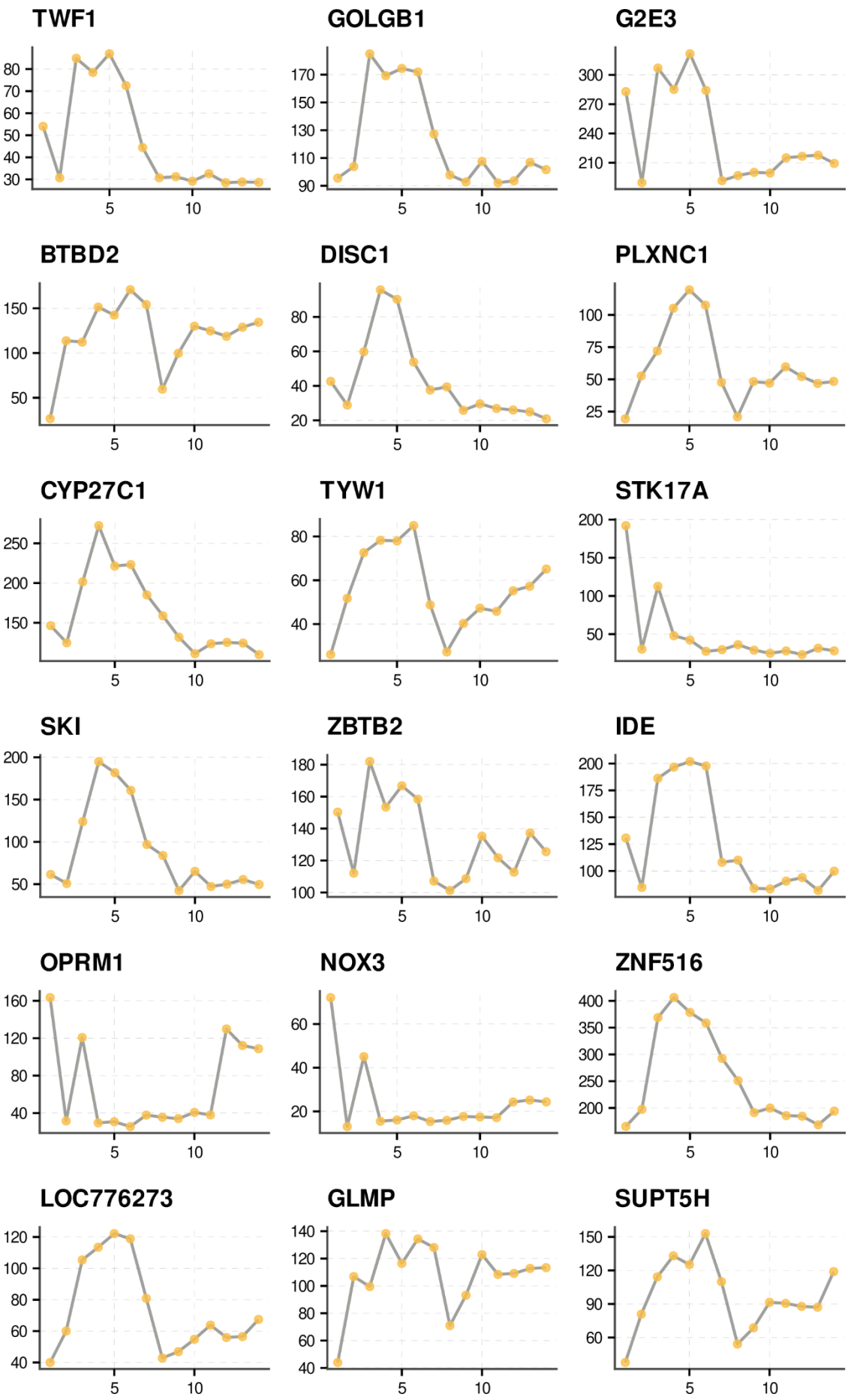

LRP12
ANAPC2
ARID1A
HECTD4
LHX9
(Gga.2348.2.S1)
PRLR
FGF13
VTN
RRBP1
YAP1
VEGFD
ANGPT2
IRF2
BMP2
HOXD13
LSAMP
CTNNA1
SMAP2

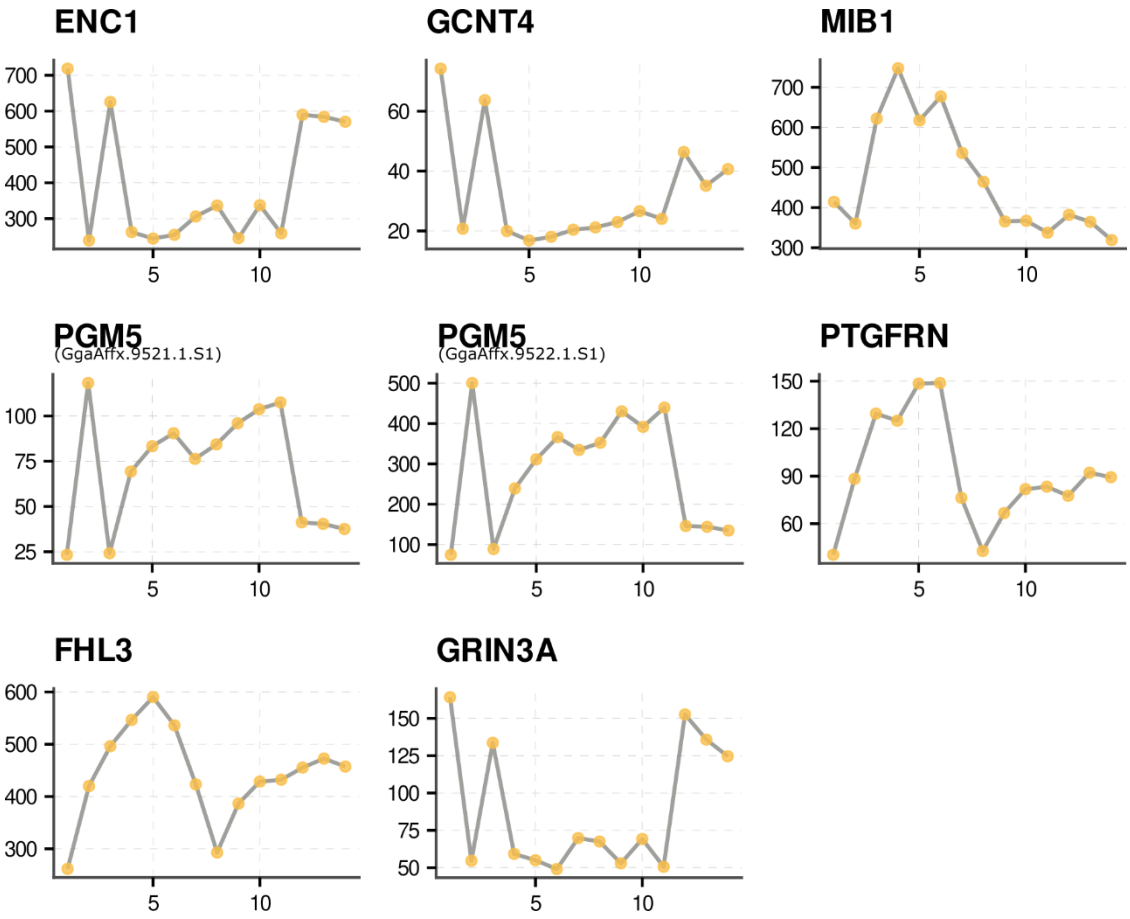

VCAN
ANO1
(Gga.6252.1.S1)
SKIL
AP1G1
GSC
EMX2
TCERG1L
EP400
ADIPOR2
FERMT1
AKAP13
(Gga.8999.1.S1)
PKP2
HOXD11
SPTBN1
(Gga.9659.1.S1_s)
MOXD1
DST
VPS36
OGT


ACTN1
CDC73
ZMYM4
GALNT9
RNF128
LOC415456
USP42
ASAP1
C9ORF58
FNBP1
FARP1
HAND2
SON
(GgaAffx.21899.1.S1)
SON
(GgaAffx.21899.3.S1_s)
TM9SF4
AKAP13
(GgaAffx.22069.1.S1_s)
KDM1A
SPTBN1
(GgaAffx.22491.2.S1_s)


SPTBN1
(GgaAffx.22491.4.S1_s)
KIF16B
SCAMP4
MIA3
JUP
MYO9B
GPD2
CDC14B
TRIO
CENPE
MAP4K4
RSF1
LIMS2
NEFL
CSNK1D
FBN3
KLHL5
TRRAP


ARID4B
LRRC8A
DNM1
CTR9
GBF1
RC3H1
TOP3A
KAT6B
SYT9
TSC2
ARHGAP29
RPS6KA6
(GgaAffx.4438.1.S1)
RPS6KA6
(GgaAffx.4438.2.S1_s)
FAM234A
ANO1
(GgaAffx.4752.1.S1_s)
ARHGAP21
MED13L
LCLAT1


TWF1
GOLGB1
G2E3
BTBD2
DISC1
PLXNC1
CYP27C1
TYW1
STK17A
SKI
ZBTB2
IDE
OPRM1
NOX3
ZNF516
LOC776273
GLMP
SUPT5H


ENC1
GCNT4
MIB1
PGM5
(GgaAffx.9521.1.S1)
PGM5
(GgaAffx.9522.1.S1)
PTGFRN
FHL3
GRIN3A

Annex 8 | List of functionally enriched GO categories

Annex 8.1 - List of functionally enriched GO categories in the PSM K1

PSM Cluster 1 Biological Process Enrichment								
	GO.ID	Term	Annotated	Significant	Expected	p-value	Proportions	Short Name
1	GO:0009653	anatomical structure morphogenesis	331	13	2,88	0,00029	0,38	anatomical morphogenesis
2	GO:0002009	morphogenesis of an epithelium	53	4	0,46	0,00036	0,12	epithelium morphogenesis
3	GO:0035295	tube development	113	5	0,98	0,00075	0,15	tube development
4	GO:0008285	negative regulation of cell population proliferation	66	4	0,58	0,00084	0,12	cell proliferation inhibition
5	GO:0007275	multicellular organism development	609	18	5,31	0,00086	0,53	multicellular organism development
6	GO:0007389	pattern specification process	72	4	0,63	0,00117	0,12	pattern specification process
7	GO:0048522	positive regulation of cellular process	522	10	4,55	0,00120	0,29	cellular process up-regulation
8	GO:0010594	regulation of endothelial cell migration	10	2	0,09	0,00186	0,06	endothelial cell migration
9	GO:0060173	limb development	38	3	0,33	0,00190	0,09	limb development
10	GO:0003170	heart valve development	11	2	0,10	0,00226	0,06	heart valve development
11	GO:0007155	cell adhesion	148	5	1,29	0,00253	0,15	cell adhesion
12	GO:0060411	cardiac septum morphogenesis	13	2	0,11	0,00319	0,06	heart septum morphogenesis
13	GO:0060429	epithelium development	112	7	0,98	0,00429	0,21	epithelium development
14	GO:0014706	striated muscle tissue development	52	3	0,45	0,00468	0,09	striated muscle development
15	GO:0050896	response to stimulus	836	16	7,29	0,00498	0,47	response to stimulus
16	GO:0014902	myotube differentiation	17	2	0,15	0,00546	0,06	myotube differentiation
17	GO:0043401	steroid hormone mediated signaling pathway	17	2	0,15	0,00546	0,06	steroid hotmone signaling
18	GO:0007417	central nervous system development	97	6	0,85	0,00579	0,18	CNS development

19	GO:0008543	fibroblast growth factor receptor signaling pathway	18	2	0,16	0,00612	0,06	FGFR signaling
20	GO:0030522	intracellular receptor signaling pathway	18	2	0,16	0,00612	0,06	intracellular signaling
21	GO:0022008	neurogenesis	191	8	1,66	0,00612	0,24	
22	GO:0031175	neuron projection development	115	4	1,00	0,00649	0,12	
23	GO:0018108	peptidyl-tyrosine phosphorylation	61	3	0,53	0,00732	0,09	
24	GO:0010604	positive regulation of macromolecule metabolic process	359	7	3,13	0,00742	0,21	
25	GO:0021537	telencephalon development	20	2	0,17	0,00754	0,06	
26	GO:0007169	transmembrane receptor protein tyrosine kinase signaling	87	5	0,76	0,00832	0,15	
27	GO:2000026	regulation of multicellular organismal development	199	5	1,73	0,00898	0,15	
28	GO:0048568	embryonic organ development	68	3	0,59	0,00989	0,09	
29	GO:0021953	central nervous system neuron differentiation	23	2	0,20	0,00992	0,06	
30	GO:0048513	animal organ development	359	11	3,13	0,01040	0,32	
31	GO:0010468	regulation of gene expression	483	8	4,21	0,01067	0,24	
32	GO:0098609	cell-cell adhesion	70	3	0,61	0,01071	0,09	
33	GO:0007519	skeletal muscle tissue development	25	2	0,22	0,01166	0,06	
34	GO:0007399	nervous system development	257	11	2,24	0,01216	0,32	
35	GO:0007165	signal transduction	518	13	4,51	0,01249	0,38	
36	GO:0060538	skeletal muscle organ development	26	2	0,23	0,01259	0,06	
37	GO:0010467	gene expression	606	9	5,28	0,01297	0,26	
38	GO:0050789	regulation of biological process	1132	21	9,87	0,01301	0,62	
39	GO:0034332	adherens junction organization	27	2	0,24	0,01354	0,06	
40	GO:0051254	positive regulation of RNA metabolic process	222	5	1,93	0,01405	0,15	
41	GO:0048646	anatomical structure formation involved in morphogenesis	144	4	1,26	0,01414	0,12	
42	GO:0050794	regulation of cellular process	1063	20	9,26	0,01476	0,59	
43	GO:0098742	cell-cell adhesion via plasma-membrane adhesion molecules	30	2	0,26	0,01657	0,06	

44	GO:0048592	eye morphogenesis	30	2	0,26	0,01657	0,06
45	GO:0022607	cellular component assembly	321	6	2,80	0,01661	0,18
46	GO:0060562	epithelial tube morphogenesis	31	2	0,27	0,01764	0,06
47	GO:0051252	regulation of RNA metabolic process	424	7	3,70	0,01789	0,21
48	GO:0023052	signaling	551	13	4,80	0,01792	0,38
49	GO:0044087	regulation of cellular component biogenesis	85	3	0,74	0,01808	0,09
50	GO:0034329	cell junction assembly	32	2	0,28	0,01874	0,06
51	GO:0051260	protein homooligomerization	32	2	0,28	0,01874	0,06
52	GO:0045935	positive regulation of nucleobase-containing compound metabolic process	240	5	2,09	0,01917	0,15
53	GO:0007154	cell communication	559	13	4,87	0,01945	0,38
54	GO:0035239	tube morphogenesis	88	3	0,77	0,01983	0,09
55	GO:0010628	positive regulation of gene expression	244	5	2,13	0,02046	0,15
56	GO:0051716	cellular response to stimulus	704	14	6,14	0,02081	0,41
57	GO:2000112	regulation of cellular macromolecule biosynthetic process	441	7	3,84	0,02185	0,21
58	GO:0044085	cellular component biogenesis	346	6	3,02	0,02332	0,18
59	GO:0010556	regulation of macromolecule biosynthetic process	450	7	3,92	0,02418	0,21
60	GO:0019219	regulation of nucleobase-containing compound metabolic process	450	7	3,92	0,02418	0,21
61	GO:0051173	positive regulation of nitrogen compound metabolic process	349	6	3,04	0,02423	0,18
62	GO:0043068	positive regulation of programmed cell death	37	2	0,32	0,02466	0,06
63	GO:0072359	circulatory system development	127	6	1,11	0,02474	0,18
64	GO:0001501	skeletal system development	97	3	0,85	0,02561	0,09
65	GO:0065007	biological regulation	1225	21	10,68	0,02564	0,62
66	GO:0006357	regulation of transcription by RNA polymerase II	262	5	2,28	0,02697	0,15
67	GO:0010942	positive regulation of cell death	39	2	0,34	0,02722	0,06

68	GO:0009952	anterior/posterior pattern specification	39	2	0,34	0,02722	0,06
69	GO:0034330	cell junction organization	39	2	0,34	0,02722	0,06
70	GO:0031325	positive regulation of cellular metabolic process	360	6	3,14	0,02779	0,18
71	GO:0031326	regulation of cellular biosynthetic process	466	7	4,06	0,02875	0,21
72	GO:0006366	transcription by RNA polymerase II	268	5	2,34	0,02941	0,15
73	GO:0051259	protein complex oligomerization	41	2	0,36	0,02987	0,06
74	GO:0051241	negative regulation of multicellular organismal process	103	3	0,90	0,02992	0,09
75	GO:0009889	regulation of biosynthetic process	470	7	4,10	0,02998	0,21
76	GO:0045595	regulation of cell differentiation	183	4	1,59	0,03119	0,12
77	GO:0097485	neuron projection guidance	43	2	0,37	0,03263	0,06
78	GO:0007411	axon guidance	43	2	0,37	0,03263	0,06
79	GO:2000765	regulation of cytoplasmic translation	5	1	0,04	0,03290	0,03
80	GO:2001014	regulation of skeletal muscle cell differentiation	5	1	0,04	0,03290	0,03
81	GO:2000738	positive regulation of stem cell differentiation	5	1	0,04	0,03290	0,03
82	GO:0110111	negative regulation of animal organ morphogenesis	5	1	0,04	0,03290	0,03
83	GO:0060193	positive regulation of lipase activity	5	1	0,04	0,03290	0,03
84	GO:0051932	synaptic transmission	5	1	0,04	0,03290	0,03
85	GO:0106030	neuron projection fasciculation	5	1	0,04	0,03290	0,03
86	GO:0036363	transforming growth factor beta activation	5	1	0,04	0,03290	0,03
87	GO:0010518	positive regulation of phospholipase activity	5	1	0,04	0,03290	0,03
88	GO:0050779	RNA destabilization	5	1	0,04	0,03290	0,03
89	GO:0035116	embryonic hindlimb morphogenesis	5	1	0,04	0,03290	0,03
90	GO:0007413	axonal fasciculation	5	1	0,04	0,03290	0,03
91	GO:0019511	peptidyl-proline hydroxylation	5	1	0,04	0,03290	0,03
92	GO:0061157	mRNA destabilization	5	1	0,04	0,03290	0,03
93	GO:0007420	brain development	67	4	0,58	0,03345	0,12

94	GO:0060485	mesenchyme development	45	2	0,39	0,03549	0,06
95	GO:0007517	muscle organ development	45	2	0,39	0,03549	0,06
96	GO:0051674	localization of cell	132	5	1,15	0,03767	0,15
97	GO:0051452	intracellular pH reduction	6	1	0,05	0,03935	0,03
98	GO:0008045	motor neuron axon guidance	6	1	0,05	0,03935	0,03
99	GO:0033627	cell adhesion mediated by integrin	6	1	0,05	0,03935	0,03
100	GO:0140253	cell-cell fusion	6	1	0,05	0,03935	0,03
101	GO:0006949	syncytium formation	6	1	0,05	0,03935	0,03
102	GO:0061014	positive regulation of mRNA catabolic process	6	1	0,05	0,03935	0,03
103	GO:0045851	pH reduction	6	1	0,05	0,03935	0,03
104	GO:0003148	outflow tract septum morphogenesis	6	1	0,05	0,03935	0,03
105	GO:0060191	regulation of lipase activity	6	1	0,05	0,03935	0,03
106	GO:0007520	myoblast fusion	6	1	0,05	0,03935	0,03
107	GO:0003281	ventricular septum development	6	1	0,05	0,03935	0,03
108	GO:0001937	negative regulation of endothelial cell proliferation	6	1	0,05	0,03935	0,03
109	GO:0010517	regulation of phospholipase activity	6	1	0,05	0,03935	0,03
110	GO:0035137	hindlimb morphogenesis	6	1	0,05	0,03935	0,03
111	GO:0000768	syncytium formation by plasma membrane fusion	6	1	0,05	0,03935	0,03
112	GO:0048566	embryonic digestive tract development	6	1	0,05	0,03935	0,03
113	GO:0010623	programmed cell death involved in cell development	6	1	0,05	0,03935	0,03
114	GO:0007507	heart development	73	5	0,64	0,03988	0,15
115	GO:0090596	sensory organ morphogenesis	48	2	0,42	0,03994	0,06
116	GO:0006355	regulation of transcription	392	6	3,42	0,04014	0,18
117	GO:1903506	regulation of nucleic acid-templated transcription	398	6	3,47	0,04280	0,18
118	GO:2001141	regulation of RNA biosynthetic process	398	6	3,47	0,04280	0,18
119	GO:0051171	regulation of nitrogen compound metabolic process	620	8	5,40	0,04297	0,24
120	GO:0044238	primary metabolic process	1175	14	10,24	0,04302	0,41

121	GO:0003151	outflow tract morphogenesis	7	1	0,06	0,04577	0,03
122	GO:2000736	regulation of stem cell differentiation	7	1	0,06	0,04577	0,03
123	GO:0008217	regulation of blood pressure	7	1	0,06	0,04577	0,03
124	GO:0015669	gas transport	7	1	0,06	0,04577	0,03
125	GO:0043550	regulation of lipid kinase activity	7	1	0,06	0,04577	0,03
126	GO:0043551	regulation of phosphatidylinositol 3-kinase activity	7	1	0,06	0,04577	0,03
127	GO:0009880	embryonic pattern specification	7	1	0,06	0,04577	0,03
128	GO:0018208	peptidyl-proline modification	7	1	0,06	0,04577	0,03
129	GO:0042572	retinol metabolic process	7	1	0,06	0,04577	0,03
130	GO:0042531	positive regulation of tyrosine phosphorylation of STAT protein	7	1	0,06	0,04577	0,03
131	GO:0021522	spinal cord motor neuron differentiation	7	1	0,06	0,04577	0,03
132	GO:0007422	peripheral nervous system development	7	1	0,06	0,04577	0,03
133	GO:0048048	embryonic eye morphogenesis	7	1	0,06	0,04577	0,03
134	GO:0032331	negative regulation of chondrocyte differentiation	7	1	0,06	0,04577	0,03
135	GO:0071300	cellular response to retinoic acid	7	1	0,06	0,04577	0,03
136	GO:0006351	transcription	406	6	3,54	0,04652	0,18
137	GO:0080090	regulation of primary metabolic process	630	8	5,49	0,04669	0,24
138	GO:0016070	RNA metabolic process	517	7	4,51	0,04731	0,21
139	GO:0009887	animal organ morphogenesis	147	5	1,28	0,04776	0,15
140	GO:0048562	embryonic organ morphogenesis	53	2	0,46	0,04782	0,06
141	GO:0032774	RNA biosynthetic process	412	6	3,59	0,04944	0,18
142	GO:0097659	nucleic acid-templated transcription	412	6	3,59	0,04944	0,18
143	GO:0048699	generation of neurons	187	7	1,63	0,04959	0,21

PSM Cluster 1 Molecular Function Enrichment								
	GO.ID	Term	Annotated	Significant	Expected	p-value	Proportions	Short Name
1	GO:0004714	transmembrane receptor protein tyrosine kinase activity	23	4	0,20	0,00001	0,13	transmembrane receptor
2	GO:0017134	fibroblast growth factor binding	7	2	0,06	0,00087	0,06	FGF binding
3	GO:0008270	zinc ion binding	74	4	0,66	0,00127	0,13	zinc ion binding
4	GO:0004879	nuclear receptor activity	9	2	0,08	0,00149	0,06	nuclear receptor
5	GO:0003707	steroid hormone receptor activity	12	2	0,11	0,00269	0,06	steroid hormone receptor
6	GO:0005102	signaling receptor binding	165	5	1,46	0,00395	0,16	signaling receptor binding
7	GO:0003674	molecular_function	1827	24	16,21	0,00438	0,75	molecular function
8	GO:0005488	binding	1478	21	13,12	0,00732	0,66	binding
9	GO:0031406	carboxylic acid binding	21	2	0,19	0,00825	0,06	carboxylic acid binding
10	GO:0008201	heparin binding	23	2	0,20	0,00986	0,06	heparin binding
11	GO:0003713	transcription coactivator activity	31	2	0,28	0,01755	0,06	transcription coactivator
12	GO:0050839	cell adhesion molecule binding	32	2	0,28	0,01865	0,06	cell adhesion molecule binding
13	GO:0001664	G protein-coupled receptor binding	32	2	0,28	0,01865	0,06	GPCR binding
14	GO:0008083	growth factor activity	34	2	0,30	0,02092	0,06	GF activity
15	GO:0048018	receptor ligand activity	93	3	0,83	0,02271	0,09	receptor ligand
16	GO:0030545	receptor regulator activity	96	3	0,85	0,02468	0,09	receptor regulator
17	GO:0005004	GPI-linked ephrin receptor activity	5	1	0,04	0,03286	0,03	GPI-linked ephrin receptor
18	GO:0019840	isoprenoid binding	5	1	0,04	0,03286	0,03	isoprenoid binding
19	GO:0031418	L-ascorbic acid binding	5	1	0,04	0,03286	0,03	L-ascorbic acid binding
20	GO:0005501	retinoid binding	5	1	0,04	0,03286	0,03	retinoid binding
21	GO:0030374	nuclear receptor transcription coactivator activity	6	1	0,05	0,03930	0,03	
22	GO:0036002	pre-mRNA binding	6	1	0,05	0,03930	0,03	
23	GO:0005184	neuropeptide hormone activity	6	1	0,05	0,03930	0,03	
24	GO:0008238	exopeptidase activity	7	1	0,06	0,04571	0,03	

PSM Cluster 1 Cellular Component Enrichment								
	GO.ID	Term	Annotated	Significant	Expected	p-value	Proportions	Short Name
1	GO:0005575	cellular component	2054	28	18,53	0,00018	0,76	cellular component
2	GO:0044421	extracellular region part	245	9	2,21	0,00040	0,24	extracellular region
3	GO:0016021	integral component of membrane	441	10	3,98	0,00041	0,27	membrane component
4	GO:0044459	plasma membrane part	249	6	2,25	0,00572	0,16	plasma membrane I
5	GO:0005604	basement membrane	18	2	0,16	0,00643	0,05	basement membrane
6	GO:0045177	apical part of cell	21	2	0,19	0,00871	0,05	apical part of cell
7	GO:0005794	Golgi apparatus	123	4	1,11	0,00903	0,11	Golgi apparatus
8	GO:0043235	receptor complex	66	3	0,60	0,00979	0,08	receptor complex
9	GO:0005887	integral component of plasma membrane	132	4	1,19	0,01153	0,11	plasma membrane II
10	GO:0031226	intrinsic component of plasma membrane	138	4	1,25	0,01341	0,11	plasma membrane III
11	GO:0005615	extracellular space	214	5	1,93	0,01356	0,14	extracellular space
12	GO:0030133	transport vesicle	30	2	0,27	0,01738	0,05	transport vesicle
13	GO:0005886	plasma membrane	430	10	3,88	0,01774	0,27	plasma membrane IV
14	GO:0071944	cell periphery	445	10	4,01	0,02311	0,27	cell periphery
15	GO:0044291	cell-cell contact zone	5	1	0,05	0,03369	0,03	cell-cell contact zone
16	GO:0031012	extracellular matrix	66	4	0,60	0,03677	0,11	extracellular matrix
17	GO:0005911	cell-cell junction	46	2	0,42	0,03868	0,05	cell-cell junction
18	GO:0005902	microvillus	6	1	0,05	0,04030	0,03	microvillus
19	GO:0043209	myelin sheath	6	1	0,05	0,04030	0,03	myelin sheath

Annex 8.2 - List of functionally enriched GO categories in the PSM K2

PSM Cluster 2 Biological Process Enrichment								
	GO.ID	Term	Annotated	Significant	Expected	p-value	Proportions	Short Name
1	GO:0015671	oxygen transport	6	4	0,02	7,70E-10	0,29	oxygen transport
2	GO:0042744	hydrogen peroxide catabolic process	7	4	0,03	1,80E-09	0,29	hydrogen peroxide catabolism
3	GO:0098869	cellular oxidant detoxification	13	4	0,05	3,60E-08	0,29	oxidant detoxification
4	GO:0070507	regulation of microtubule cytoskeleton organization	21	2	0,08	0,00180	0,14	microtubule dynamics
5	GO:0006355	regulation of transcription	392	5	1,41	0,00440	0,36	transcription regulation
6	GO:0051130	positive regulation of cellular component organization	116	3	0,42	0,00460	0,21	up regulation of cell organization
7	GO:0007275	multicellular organism development	609	6	2,19	0,00560	0,43	multicellular organism development
8	GO:0030154	cell differentiation	448	5	1,61	0,00780	0,36	cell differentiation
9	GO:0051640	organelle localization	48	2	0,17	0,00900	0,14	organelle localization
10	GO:0010638	positive regulation of organelle organization	57	2	0,20	0,01260	0,14	organelle organization
11	GO:0051412	response to corticosterone	5	1	0,02	0,01530	0,07	corticosterone response
12	GO:0051385	response to mineralocorticoid	5	1	0,02	0,01530	0,07	mineralocorticoid response
13	GO:0051294	establishment of spindle orientation	5	1	0,02	0,01530	0,07	mitotic spindle orientation I
14	GO:0045069	regulation of viral genome replication	5	1	0,02	0,01530	0,07	viral genome replication I
15	GO:0072698	protein localization to microtubule cytoskeleton	5	1	0,02	0,01530	0,07	microtubule localization
16	GO:0044380	protein localization to cytoskeleton	5	1	0,02	0,01530	0,07	cytoskeleton localization
17	GO:0000132	establishment of mitotic spindle orientation	5	1	0,02	0,01530	0,07	mitotic spindle orientation II
18	GO:0019079	viral genome replication	5	1	0,02	0,01530	0,07	viral genome replication II
19	GO:0051450	myoblast proliferation	5	1	0,02	0,01530	0,07	myoblast proliferation

20	GO:0051293	establishment of spindle localization	6	1	0,02	0,01830	0,07	mitotic spindle localization
21	GO:0046605	regulation of centrosome cycle	6	1	0,02	0,01830	0,07	
22	GO:0040001	establishment of mitotic spindle localization	6	1	0,02	0,01830	0,07	
23	GO:0031468	nuclear envelope reassembly	6	1	0,02	0,01830	0,07	
24	GO:0051653	spindle localization	7	1	0,03	0,02140	0,07	
25	GO:0006998	nuclear envelope organization	7	1	0,03	0,02140	0,07	
26	GO:0033993	response to lipid	77	2	0,28	0,02230	0,14	
27	GO:1903900	regulation of viral life cycle	8	1	0,03	0,02440	0,07	
28	GO:0045840	positive regulation of mitotic nuclear division	8	1	0,03	0,02440	0,07	
29	GO:0031935	regulation of chromatin silencing	8	1	0,03	0,02440	0,07	
30	GO:0001947	heart looping	8	1	0,03	0,02440	0,07	
31	GO:0061371	determination of heart left/right asymmetry	8	1	0,03	0,02440	0,07	
32	GO:0043901	negative regulation of multi-organism process	9	1	0,03	0,02740	0,07	
33	GO:0061515	myeloid cell development	9	1	0,03	0,02740	0,07	
34	GO:0003143	embryonic heart tube morphogenesis	9	1	0,03	0,02740	0,07	
35	GO:0035050	embryonic heart tube development	9	1	0,03	0,02740	0,07	
36	GO:0045663	positive regulation of myoblast differentiation	9	1	0,03	0,02740	0,07	
37	GO:0032418	lysosome localization	9	1	0,03	0,02740	0,07	
38	GO:0006997	nucleus organization	10	1	0,04	0,03040	0,07	
39	GO:0060968	regulation of gene silencing	11	1	0,04	0,03340	0,07	
40	GO:0051785	positive regulation of nuclear division	11	1	0,04	0,03340	0,07	
41	GO:0040019	positive regulation of embryonic development	11	1	0,04	0,03340	0,07	
42	GO:0042446	hormone biosynthetic process	11	1	0,04	0,03340	0,07	
43	GO:0019058	viral life cycle	11	1	0,04	0,03340	0,07	
44	GO:0007098	centrosome cycle	11	1	0,04	0,03340	0,07	
45	GO:0050792	regulation of viral process	11	1	0,04	0,03340	0,07	
46	GO:0007368	determination of left/right symmetry	12	1	0,04	0,03630	0,07	
47	GO:0050772	positive regulation of axonogenesis	12	1	0,04	0,03630	0,07	
48	GO:0097305	response to alcohol	12	1	0,04	0,03630	0,07	

49	GO:0018958	phenol-containing compound metabolic process	12	1	0,04	0,03630	0,07
50	GO:0051384	response to glucocorticoid	12	1	0,04	0,03630	0,07
51	GO:0048469	cell maturation	12	1	0,04	0,03630	0,07
52	GO:0031960	response to corticosteroid	12	1	0,04	0,03630	0,07
53	GO:0043903	regulation of symbiosis	12	1	0,04	0,03630	0,07
54	GO:0031023	microtubule organizing center organization	12	1	0,04	0,03630	0,07
55	GO:0051172	negative regulation of nitrogen compound metabolic process	255	3	0,92	0,03910	0,21
56	GO:0007030	Golgi organization	13	1	0,05	0,03930	0,07
57	GO:0006342	chromatin silencing	13	1	0,05	0,03930	0,07
58	GO:1905269	positive regulation of chromatin organization	13	1	0,05	0,03930	0,07
59	GO:0007088	regulation of mitotic nuclear division	13	1	0,05	0,03930	0,07
60	GO:0045597	positive regulation of cell differentiation	107	2	0,38	0,04110	0,14
61	GO:0006357	regulation of transcription by RNA polymerase II	262	3	0,94	0,04190	0,21
62	GO:0044237	cellular metabolic process	1217	11	4,37	0,04210	0,79
63	GO:0030705	cytoskeleton-dependent intracellular transport	14	1	0,05	0,04230	0,07
64	GO:1905037	autophagosome organization	14	1	0,05	0,04230	0,07
65	GO:0048813	dendrite morphogenesis	14	1	0,05	0,04230	0,07
66	GO:0061053	somite development	14	1	0,05	0,04230	0,07
67	GO:0045814	negative regulation of gene expression	14	1	0,05	0,04230	0,07
68	GO:0030010	establishment of cell polarity	14	1	0,05	0,04230	0,07
69	GO:0045661	regulation of myoblast differentiation	14	1	0,05	0,04230	0,07
70	GO:0000045	autophagosome assembly	14	1	0,05	0,04230	0,07
71	GO:0006366	transcription by RNA polymerase II	268	3	0,96	0,04440	0,21
72	GO:0031324	negative regulation of cellular metabolic process	270	3	0,97	0,04520	0,21
73	GO:1902850	microtubule cytoskeleton organization involved in mitosis	15	1	0,05	0,04520	0,07
74	GO:0009799	specification of symmetry	15	1	0,05	0,04520	0,07

75	GO:1903828	negative regulation of cellular protein localization	15	1	0,05	0,04520	0,07	
76	GO:0061025	membrane fusion	15	1	0,05	0,04520	0,07	
77	GO:0009855	determination of bilateral symmetry	15	1	0,05	0,04520	0,07	
78	GO:0010605	negative regulation of macromolecule metabolic process	272	3	0,98	0,04610	0,21	
79	GO:0032526	response to retinoic acid	16	1	0,06	0,04820	0,07	

PSM Cluster 2 Molecular Function Enrichment								
	GO.ID	Term	Annotated	Significant	Expected	p-value	Proportions	Short Name
1	GO:0031720	haptoglobin binding	5	4	0,02	2,30E-10	0,31	haptoglobin binding
2	GO:0005344	oxygen carrier activity	6	4	0,02	7,00E-10	0,31	oxygen carrier
3	GO:0004601	peroxidase activity	8	4	0,03	3,30E-09	0,31	peroxidase
4	GO:0019825	oxygen binding	9	4	0,03	5,90E-09	0,31	oxygen binding
5	GO:0020037	heme binding	24	4	0,09	4,80E-07	0,31	heme binding
6	GO:0043177	organic acid binding	26	4	0,09	6,80E-07	0,31	organic acid binding
7	GO:0005506	iron ion binding	25	3	0,09	0,00005	0,23	iron ion binding
8	GO:0001085	RNA polymerase II transcription factor binding	25	2	0,09	0,00240	0,15	RNA Pol II binding
9	GO:0005515	protein binding	780	10	2,81	0,00800	0,77	protein binding
10	GO:0003700	DNA-binding transcription factor activity	162	3	0,58	0,01130	0,23	DNA binding
11	GO:0015459	potassium channel regulator activity	6	1	0,02	0,01820	0,08	potassium channel regulator
12	GO:0140110	transcription regulator activity	201	3	0,72	0,02020	0,23	transcription regulator
13	GO:0000987	cis-regulatory region sequence-specific DNA binding	76	2	0,27	0,02130	0,15	DNA binding region
14	GO:0000978	RNA polymerase II cis-regulatory region sequence-specific DNA binding	76	2	0,27	0,02130	0,15	RNA Pol II DNA binding

15	GO:0043130	ubiquitin binding	8	1	0,03	0,02420	0,08	ubiquitin binding
16	GO:0099106	ion channel regulator activity	8	1	0,03	0,02420	0,08	ion channel regulator
17	GO:0032182	ubiquitin-like protein binding	10	1	0,04	0,03010	0,08	ubiquitin-like protein binding
18	GO:0016247	channel regulator activity	11	1	0,04	0,03310	0,08	channel regulator
19	GO:0002020	protease binding	11	1	0,04	0,03310	0,08	protease binding
20	GO:0004867	serine-type endopeptidase inhibitor activity	14	1	0,05	0,04190	0,08	endopeptidase inhibitor
21	GO:0000977	RNA polymerase II transcription regulatory region sequence-specific DNA binding	119	2	0,43	0,04890	0,15	
22	GO:0001012	RNA polymerase II transcription regulatory region sequence-specific DNA binding	120	2	0,43	0,04960	0,15	

PSM Cluster 2 Cellular Component Enrichment

	GO.ID	Term	Annotated	Significant	Expected	p-value	Proportions	Short Name
1	GO:0031838	haptoglobin-hemoglobin complex	5	4	0,02	3,00E-10	0,27	haptoglobin-hemoglobin complex
2	GO:0005833	hemoglobin complex	5	4	0,02	3,00E-10	0,27	hemoglobin complex
3	GO:0045111	intermediate filament cytoskeleton	19	2	0,07	0,00150	0,13	intermediate filament I
4	GO:0005575	cellular_component	2054	13	7,51	0,00750	0,87	cellular component
5	GO:0005801	cis-Golgi network	5	1	0,02	0,01580	0,07	cis-Golgi network
6	GO:0044430	cytoskeletal part	186	3	0,68	0,01880	0,20	cytoskeleton
7	GO:0099513	polymeric cytoskeletal fiber	78	2	0,29	0,02430	0,13	cytoskeletal fiber
8	GO:0044431	Golgi apparatus part	81	2	0,30	0,02610	0,13	Golgi apparatus
9	GO:0005795	Golgi stack	11	1	0,04	0,03440	0,07	Golgi stack
10	GO:0005798	Golgi-associated vesicle	13	1	0,05	0,04050	0,07	Golgi-associated vesicle
11	GO:0015630	microtubule cytoskeleton	111	2	0,41	0,04660	0,13	microtubule
12	GO:0005882	intermediate filament	16	1	0,06	0,04960	0,07	intermediate filament II

Annex 8.3 - List of functionally enriched GO categories in the Limb K1

Limb Biological Process Enrichment								
	GO.ID	Term	Annotated	Significant	Expected	p-value	Proportions	Short Name
1	GO:0007275	multicellular organism development	609	16	4,68	0,00001	0,53	multicellular organism development
2	GO:0006368	transcription elongation from RNA polymerase II promoter	5	2	0,04	0,00053	0,07	RNA elongation
3	GO:0032784	regulation of DNA-dependent transcription	6	2	0,05	0,00079	0,07	elongation
4	GO:0045944	positive regulation of transcription by RNA polymerase II	164	6	1,26	0,00107	0,20	transcription activation
5	GO:0045596	negative regulation of cell differentiation	69	4	0,53	0,00153	0,13	cell differentiation inhibition
6	GO:0071495	cellular response to endogenous stimulus	148	7	1,14	0,00198	0,23	endogenous stimulus response
7	GO:0071310	cellular response to organic substance	216	8	1,66	0,00216	0,27	organic substance response
8	GO:0048468	cell development	258	7	1,98	0,00226	0,23	cell development
9	GO:0060317	cardiac epithelial to mesenchymal transition	12	2	0,09	0,00336	0,07	cardiac EMT
10	GO:0030510	regulation of BMP signaling pathway	14	2	0,11	0,00460	0,07	regulation of BMP pathway
11	GO:0000122	negative regulation of transcription by RNA polymerase II	94	4	0,72	0,00473	0,13	transcription inhibition
12	GO:0031503	protein-containing complex localization	15	2	0,12	0,00528	0,07	protein-containing complex localization
13	GO:0006915	apoptotic process	165	5	1,27	0,00663	0,17	apoptotic process
14	GO:0045597	positive regulation of cell differentiation	107	4	0,82	0,00749	0,13	activation of cell differentiation
15	GO:0032496	response to lipopolysaccharide	18	2	0,14	0,00758	0,07	response to lipopolysaccharide
16	GO:0007166	cell surface receptor signaling pathway	279	8	2,15	0,00830	0,27	cell surface receptor pathway
17	GO:0007399	nervous system development	257	6	1,98	0,01011	0,20	nervous system development

18	GO:0090100	positive regulation of transmembrane receptor protein serine/threonine kinase signaling pathway	21	2	0,16	0,01027	0,07	transmembrane signaling up-regulation
19	GO:0032870	cellular response to hormone stimulus	63	3	0,48	0,01088	0,10	hormone stimulus response
20	GO:0071363	cellular response to growth factor stimulus	84	5	0,65	0,01201	0,17	GF stimulus response
21	GO:0070848	response to growth factor	87	5	0,67	0,01346	0,17	
22	GO:0048666	neuron development	132	4	1,02	0,01543	0,13	
23	GO:0048731	system development	505	10	3,88	0,01554	0,33	
24	GO:0035108	limb morphogenesis	27	2	0,21	0,01670	0,07	
25	GO:0008150	biological_process	1970	29	15,15	0,01685	0,97	
26	GO:0035107	appendage morphogenesis	28	2	0,22	0,01790	0,07	
27	GO:0030030	cell projection organization	139	4	1,07	0,01835	0,13	
28	GO:0120036	plasma membrane bounded cell projection organization	139	4	1,07	0,01835	0,13	
29	GO:0065007	biological regulation	1225	22	9,42	0,01840	0,73	
30	GO:0090068	positive regulation of cell cycle process	29	2	0,22	0,01914	0,07	
31	GO:0048646	anatomical structure formation involved in morphogenesis	144	4	1,11	0,02063	0,13	
32	GO:0009725	response to hormone	91	3	0,70	0,02896	0,10	
33	GO:0060173	limb development	38	2	0,29	0,03180	0,07	
34	GO:0048736	appendage development	39	2	0,30	0,03337	0,07	
35	GO:0009952	anterior/posterior pattern specification	39	2	0,30	0,03337	0,07	
36	GO:0001501	skeletal system development	97	3	0,75	0,03412	0,10	
37	GO:0030182	neuron differentiation	169	4	1,30	0,03465	0,13	
38	GO:0045787	positive regulation of cell cycle	40	2	0,31	0,03496	0,07	
39	GO:0071396	cellular response to lipid	41	2	0,32	0,03659	0,07	
40	GO:0072698	protein localization to microtubule cytoskeleton	5	1	0,04	0,03664	0,03	
41	GO:0043392	negative regulation of DNA binding	5	1	0,04	0,03664	0,03	
42	GO:0090504	epiboly	5	1	0,04	0,03664	0,03	
43	GO:0090505	epiboly involved in wound healing	5	1	0,04	0,03664	0,03	
44	GO:0044380	protein localization to cytoskeleton	5	1	0,04	0,03664	0,03	

45	GO:0031113	regulation of microtubule polymerization	5	1	0,04	0,03664	0,03
46	GO:0031115	negative regulation of microtubule polymerization	5	1	0,04	0,03664	0,03
47	GO:0044319	wound healing	5	1	0,04	0,03664	0,03
48	GO:0003198	epithelial to mesenchymal transition involved in endocardial cushion formation	5	1	0,04	0,03664	0,03
49	GO:0060795	cell fate commitment involved in formation of primary germ layer	5	1	0,04	0,03664	0,03
50	GO:0031623	receptor internalization	5	1	0,04	0,03664	0,03
51	GO:0035116	embryonic hindlimb morphogenesis	5	1	0,04	0,03664	0,03
52	GO:0051591	response to cAMP	5	1	0,04	0,03664	0,03
53	GO:0046683	response to organophosphorus	5	1	0,04	0,03664	0,03
54	GO:0010390	histone monoubiquitination	5	1	0,04	0,03664	0,03
55	GO:0045684	positive regulation of epidermis development	5	1	0,04	0,03664	0,03
56	GO:0060174	limb bud formation	5	1	0,04	0,03664	0,03
57	GO:0035987	endodermal cell differentiation	5	1	0,04	0,03664	0,03
58	GO:0051702	interaction with symbiont	5	1	0,04	0,03664	0,03
59	GO:0030857	negative regulation of epithelial cell differentiation	5	1	0,04	0,03664	0,03
60	GO:0051241	negative regulation of multicellular organismal process	103	3	0,79	0,03973	0,10
61	GO:0007265	Ras protein signal transduction	44	2	0,34	0,04164	0,07
62	GO:1903507	negative regulation of nucleic acid-templated transcription	145	6	1,12	0,04339	0,20
63	GO:1902679	negative regulation of RNA biosynthetic process	145	6	1,12	0,04339	0,20
64	GO:0009987	cellular process	1781	25	13,70	0,04366	0,83
65	GO:0051726	regulation of cell cycle	107	3	0,82	0,04372	0,10
66	GO:0048524	positive regulation of viral process	6	1	0,05	0,04381	0,03
67	GO:2000134	negative regulation of G1/S transition of mitotic cell cycle	6	1	0,05	0,04381	0,03

68	GO:1902807	negative regulation of cell cycle G1/S phase transition	6	1	0,05	0,04381	0,03	
69	GO:0006220	pyrimidine nucleotide metabolic process	6	1	0,05	0,04381	0,03	
70	GO:1902808	positive regulation of cell cycle G1/S phase transition	6	1	0,05	0,04381	0,03	
71	GO:0006221	pyrimidine nucleotide biosynthetic process	6	1	0,05	0,04381	0,03	
72	GO:0031111	negative regulation of microtubule polymerization or depolymerization	6	1	0,05	0,04381	0,03	
73	GO:0030216	keratinocyte differentiation	6	1	0,05	0,04381	0,03	
74	GO:0006576	cellular biogenic amine metabolic process	6	1	0,05	0,04381	0,03	
75	GO:0035137	hindlimb morphogenesis	6	1	0,05	0,04381	0,03	
76	GO:0003148	outflow tract septum morphogenesis	6	1	0,05	0,04381	0,03	
77	GO:0007019	microtubule depolymerization	6	1	0,05	0,04381	0,03	
78	GO:0001706	endoderm formation	6	1	0,05	0,04381	0,03	
79	GO:0071480	cellular response to gamma radiation	6	1	0,05	0,04381	0,03	
80	GO:0040020	regulation of meiotic nuclear division	6	1	0,05	0,04381	0,03	
81	GO:0031440	regulation of mRNA 3'-end processing	6	1	0,05	0,04381	0,03	
82	GO:0072528	pyrimidine-containing compound biosynthetic process	6	1	0,05	0,04381	0,03	
83	GO:0003272	endocardial cushion formation	6	1	0,05	0,04381	0,03	
84	GO:0016574	histone ubiquitination	6	1	0,05	0,04381	0,03	
85	GO:1901881	positive regulation of protein depolymerization	6	1	0,05	0,04381	0,03	
86	GO:0019438	aromatic compound biosynthetic process	486	14	3,74	0,04600	0,47	
87	GO:0048699	generation of neurons	187	4	1,44	0,04754	0,13	
88	GO:0009314	response to radiation	48	2	0,37	0,04876	0,07	
89	GO:0008283	cell proliferation	189	4	1,45	0,04912	0,13	

Limb Molecular Function Enrichment								
	GO.ID	Term	Annotated	Significant	Expected	p-value	Proportions	Short Name
1	GO:0043565	sequence-specific DNA binding	192	8	1,49	0,00005	0,29	DNA binding I
2	GO:0005488	binding	1478	26	11,48	0,00014	0,93	binding
3	GO:0000981	DNA-binding transcription factor activity, RNA polymerase II-specific	107	5	0,83	0,00100	0,18	DNA binding TF
4	GO:0044212	transcription regulatory region DNA binding	148	5	1,15	0,00423	0,18	DNA binding region I
5	GO:0005515	protein binding	780	14	6,06	0,00447	0,50	protein binding
6	GO:0019904	protein domain specific binding	52	3	0,40	0,00648	0,11	protein domain binding
7	GO:0003714	transcription corepressor activity	20	2	0,16	0,00944	0,07	transcription corepressor
8	GO:0000977	RNA polymerase II transcription regulatory region sequence-specific DNA binding	119	4	0,92	0,01099	0,14	RNA Pol II DNA binding region I
9	GO:0001012	RNA polymerase II transcription regulatory region sequence-specific DNA binding	120	4	0,93	0,01132	0,14	RNA Pol II DNA binding region II
10	GO:0000976	transcription regulatory region sequence-specific DNA binding	128	4	0,99	0,01411	0,14	DNA binding region II
11	GO:1990837	sequence-specific double-stranded DNA binding	133	4	1,03	0,01606	0,14	DNA binding II
12	GO:0046872	metal ion binding	460	8	3,57	0,01608	0,29	metal ion binding
13	GO:0043169	cation binding	474	8	3,68	0,01906	0,29	cation binding
14	GO:0003713	transcription coactivator activity	31	2	0,24	0,02197	0,07	transcription coactivator
15	GO:0008134	transcription factor binding	84	3	0,65	0,02384	0,11	transcription factor binding
16	GO:0003690	double-stranded DNA binding	152	4	1,18	0,02500	0,14	DNA binding III
17	GO:0003712	transcription coregulator activity	57	4	0,44	0,02678	0,14	transcription coregulator
18	GO:0004888	transmembrane signaling receptor activity	93	3	0,72	0,03105	0,11	transmembrane signaling receptor
19	GO:0005088	Ras guanyl-nucleotide exchange factor activity	5	1	0,04	0,03690	0,04	Ras activity

20	GO:0000993	RNA polymerase II complex binding	6	1	0,05	0,04412	0,04	RNA Pol II binding
21	GO:0005507	copper ion binding	6	1	0,05	0,04412	0,04	
22	GO:0001228	DNA-binding transcription activator activity, RNA polymerase II-specific	48	2	0,37	0,04931	0,07	

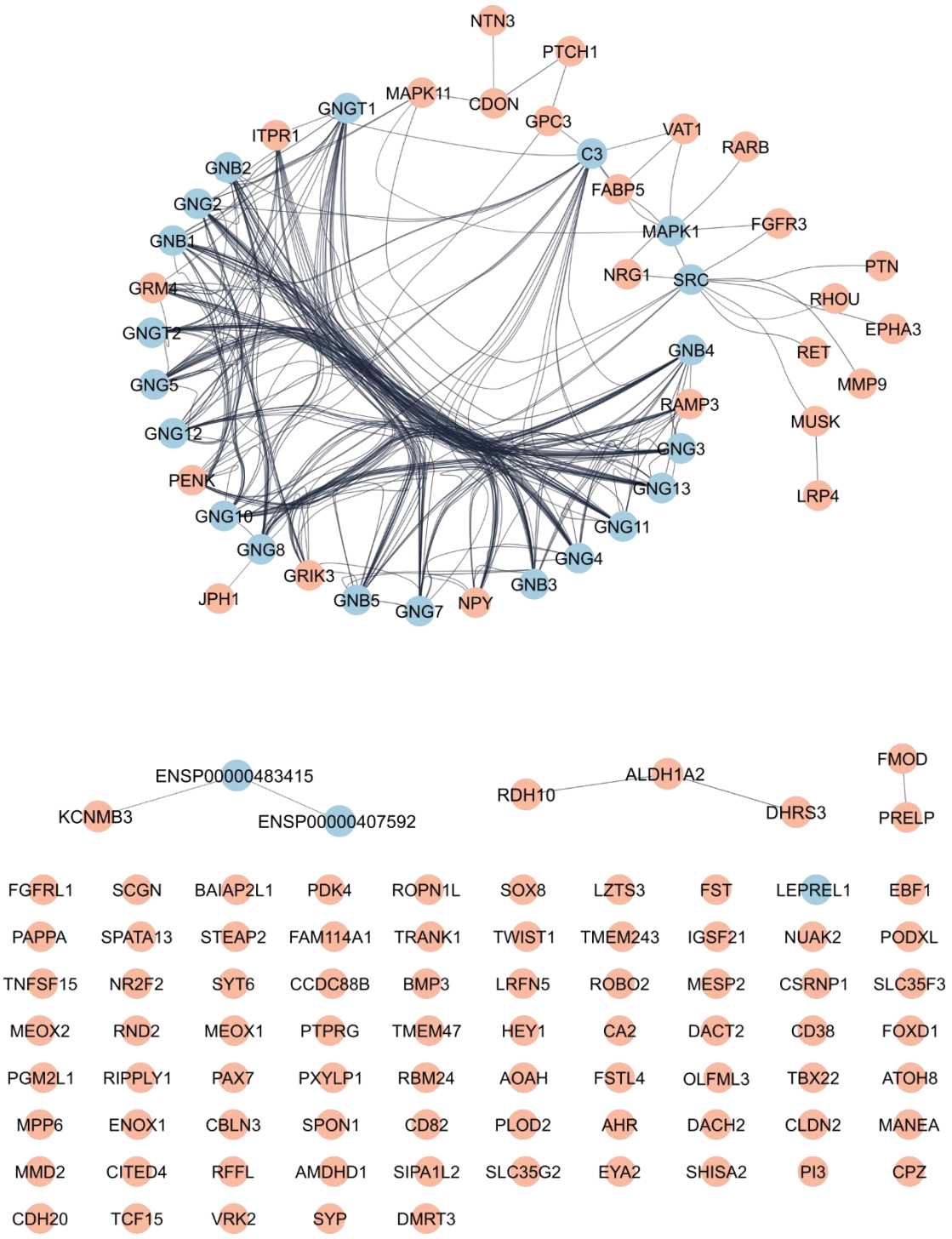
Limb Cellular Component Enrichment

	GO.ID	Term	Annotated	Significant	Expected	p-value	Proportions	Short Name
1	GO:0005575	cellular component	2054	28	14,52	0,00035	0,97	cellular component
2	GO:0044459	plasma membrane part	249	7	1,76	0,00110	0,24	plasma membrane I
3	GO:0008023	transcription elongation factor complex	8	2	0,06	0,00123	0,07	transcription elongation
4	GO:0043229	intracellular organelle	1383	20	9,78	0,00733	0,69	intracellular organelle I
5	GO:0005634	nucleus	825	13	5,83	0,00761	0,45	nucleus
6	GO:0005925	focal adhesion	22	2	0,16	0,00955	0,07	focal adhesion
7	GO:0043235	receptor complex	66	3	0,47	0,00979	0,10	receptor complex
8	GO:0030027	lamellipodium	26	2	0,18	0,01321	0,07	lamellipodium
9	GO:0044464	cell part	1831	26	12,95	0,01349	0,90	cell part
10	GO:0005623	cell	1848	26	13,07	0,01572	0,90	cell
11	GO:0044444	cytoplasmic part	830	11	5,87	0,01589	0,38	cytoplasmic part
12	GO:0005737	cytoplasm	1199	14	8,48	0,01621	0,48	cytoplasm
13	GO:0030659	cytoplasmic vesicle membrane	40	2	0,28	0,02989	0,07	cytoplasmic vesicle
14	GO:0012506	vesicle membrane	42	2	0,30	0,03272	0,07	vesicle
15	GO:0098636	protein complex involved in cell adhesion	5	1	0,04	0,03369	0,03	cell adhesion complex
16	GO:0097433	dense body	5	1	0,04	0,03369	0,03	dense body
17	GO:0030136	clathrin-coated vesicle	5	1	0,04	0,03369	0,03	clathrin-coated vesicle
18	GO:0031224	intrinsic component of membrane	468	7	3,31	0,03380	0,24	intrinsic membrane
19	GO:0044446	intracellular organelle part	885	12	6,26	0,03575	0,41	intracellular organelle II
20	GO:0098552	side of membrane	45	2	0,32	0,03715	0,07	membrane
21	GO:0031252	cell leading edge	46	2	0,33	0,03868	0,07	
22	GO:0005905	clathrin-coated pit	6	1	0,04	0,04030	0,03	

23	GO:0030120	vesicle coat	6	1	0,04	0,04030	0,03
24	GO:0031981	nuclear lumen	315	7	2,23	0,04092	0,24
25	GO:0044463	cell projection part	117	3	0,83	0,04416	0,10
26	GO:0120038	plasma membrane bounded cell projection	117	3	0,83	0,04416	0,10
27	GO:0044422	organelle part	916	12	6,48	0,04446	0,41
28	GO:0016328	lateral plasma membrane	7	1	0,05	0,04686	0,03
29	GO:0030667	secretory granule membrane	7	1	0,05	0,04686	0,03
30	GO:0005923	bicellular tight junction	7	1	0,05	0,04686	0,03
31	GO:0070160	tight junction	7	1	0,05	0,04686	0,03
32	GO:0120025	plasma membrane bounded cell projection	204	4	1,44	0,04771	0,14
33	GO:0005654	nucleoplasm	230	6	1,63	0,04923	0,21

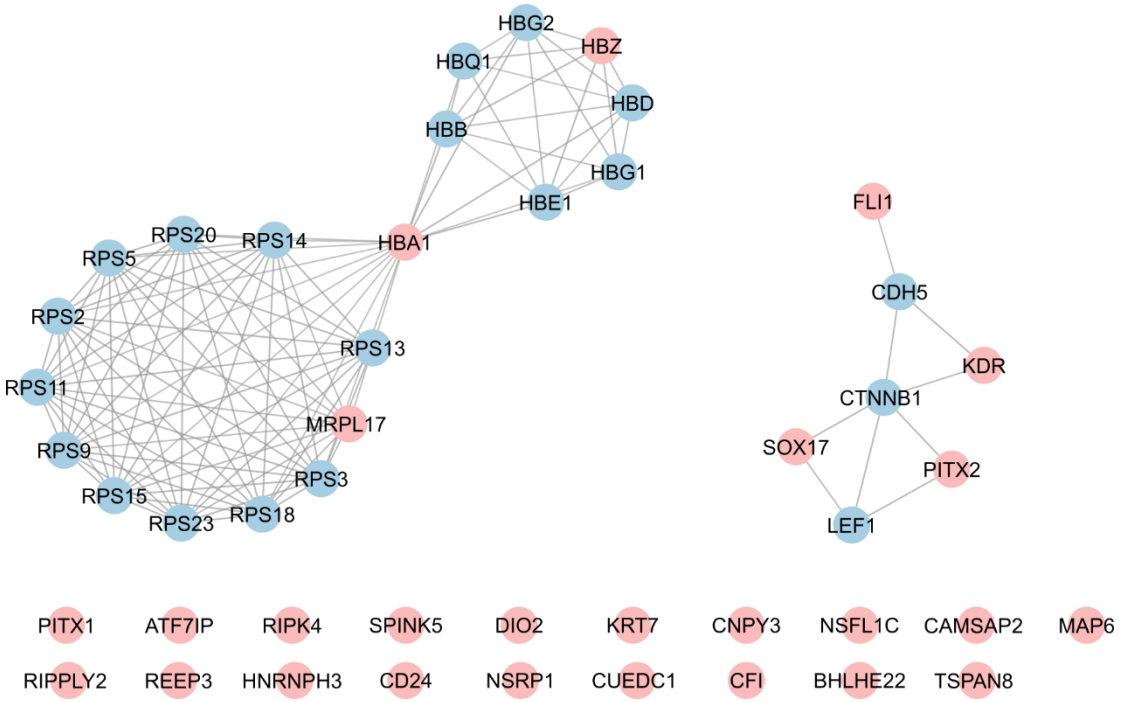
Annex 9 | Functional Interaction network before MCL clustering

Annex 9.1 - Functional Interaction network based on PSM K1 genes



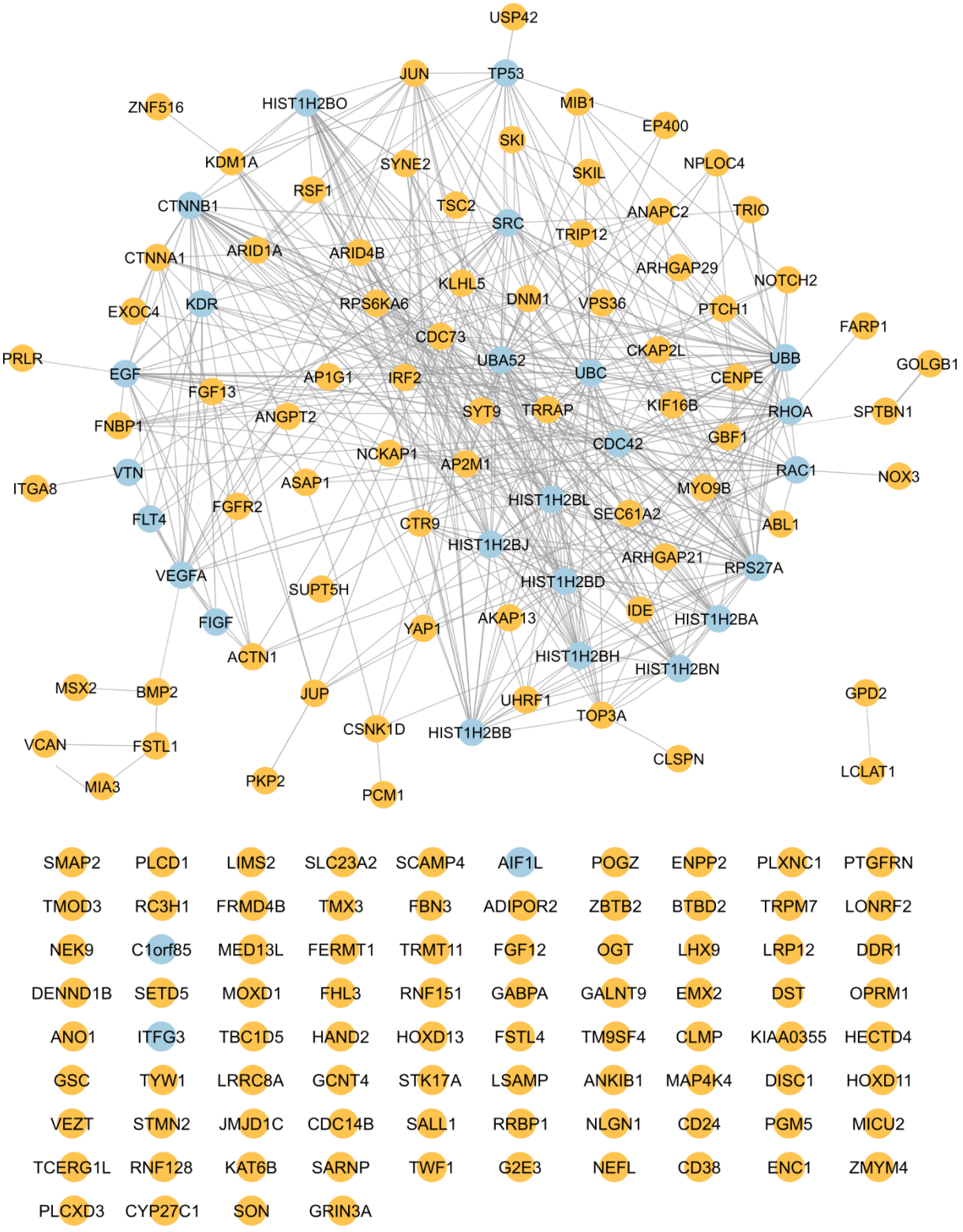
Original Functional Interaction network visualization for proteins coded by the genes from the PSM K1, with 20 additional predicted interactors, before the MCL clustering. One module was found before MCL Clustering with 44 nodes and 294 edges. Pink – ClockOME interactors; Blue – additional predicted interactors;

Annex 9.2 - Functional Interaction network based on PSM K2 genes



Original Functional Interaction network visualization for proteins coded by the genes from the PSM K2, with 20 additional predicted interactors, before the MCL clustering. Two module were found before MCL Clustering. The left network is composed of 20 nodes and 103 edges; the righ network is composed of 7 nodes and 9 edges. Pink – ClockOME interactors; Blue – additional predicted interactors.

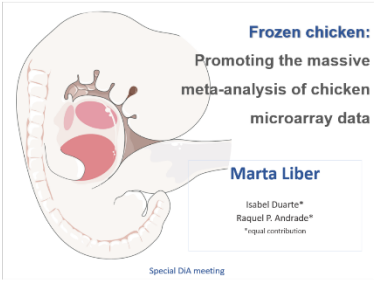
Annex 9.3 - Functional Interaction network based on Limb K1 genes

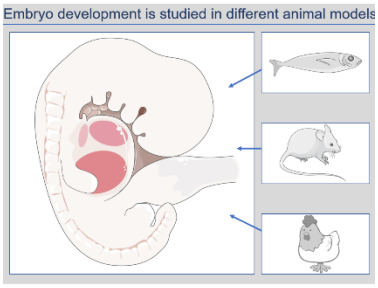


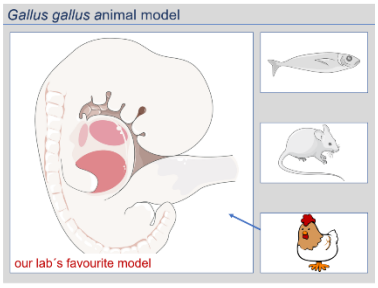
Original Functional Interaction network visualization for proteins coded by the genes from the PSM K2, with 20 additional predicted interactors, before the MCL clustering. Two module were found before MCL Clustering. The left network is composed of 20 nodes and 103 edges; the righ network is composed of 7 nodes and 9 edges. Pink – ClockOME interactors; Blue – additional predicted interactors.

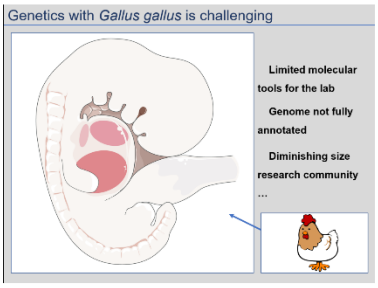
Annex 10 | Masters Portfolio – Selected Talks

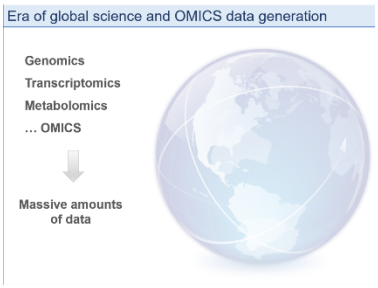
Special DiA Meeting | June 2019

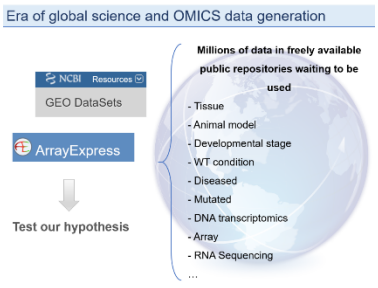


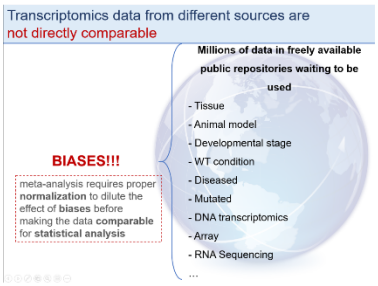


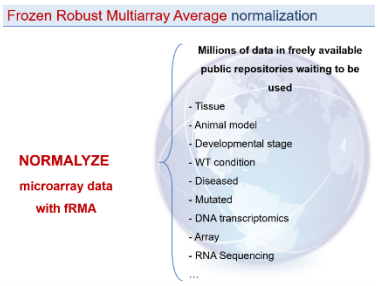


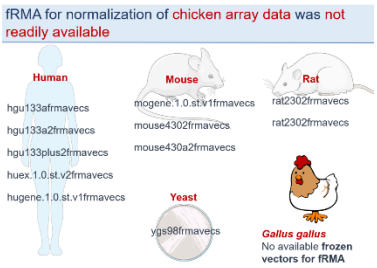


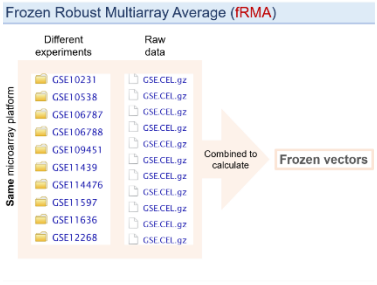


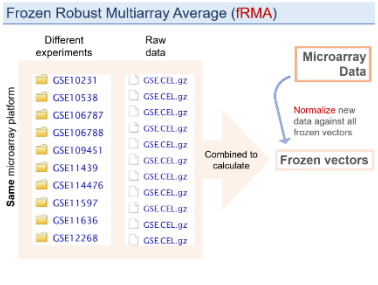


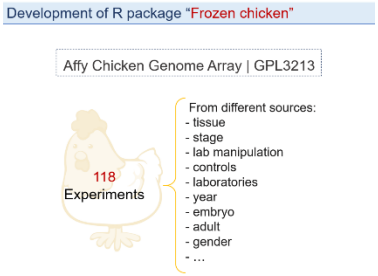


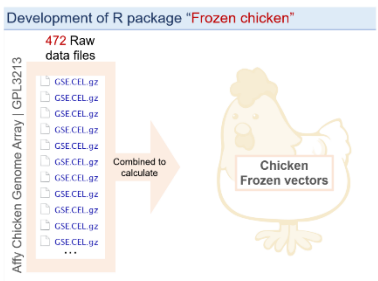


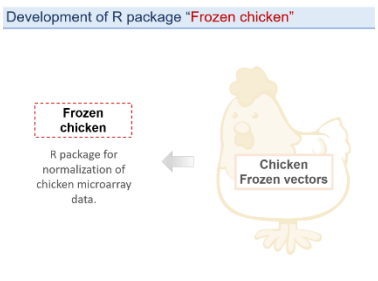


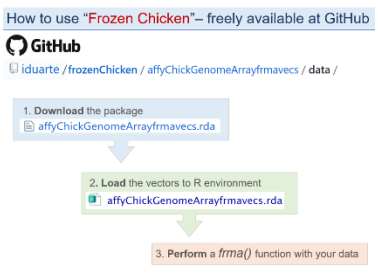




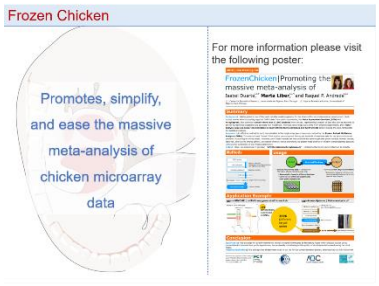








Special DiA Meeting | June 2019



Annex 11 | Masters Portfolio – "Frozen Chicken" Selected Poster

2019 | DiA Meeting Far

FrozenChicken|Promoting the massive meta-analysis of

Isabel Duarte,[1,2] **Marta Liber**,[1,2] and Raquel P. Andrade[1,2,3]

(1) Centre for Biomedical Research, Universidade do Algarve, Faro, Portugal (2) Algarve Biomedical Centre, Universidade of Algarve, Faro, Portugal

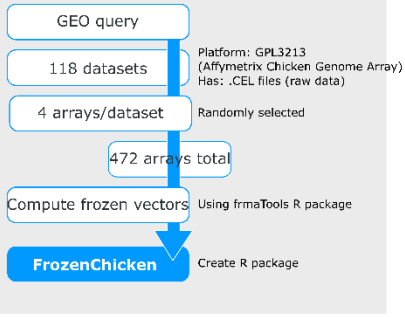
Summary

Background | *Gallus gallus* is one of the most valuable model organisms for the study of the vertebrate embryo development. Such studies can be aided by pooling together OMICs data from public repositories, like **Gene Expression Omnibus (GEO)** and **ArrayExpress,** that currently **contain more than 11.660 datasets** from chicken, representing a wealth of data that can be explored to answer fundamental questions and generate new hypothesis. However, since these data come from different experiments, their **meta-analysis requires proper normalization to deal with the technical biases and batch effects** before making the data comparable for statistical analysis.

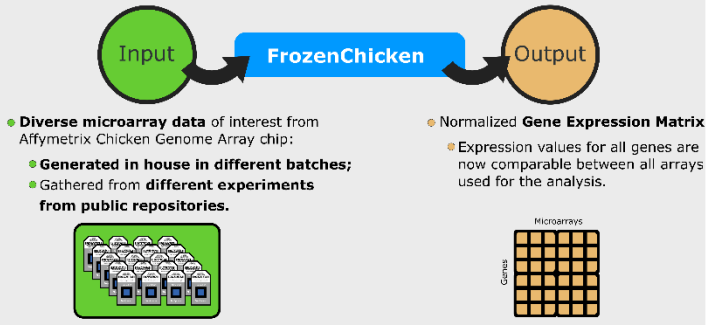
Approach | An effective method for such normalization is the single-array pre-processing provided by the **Frozen Robust Multiarray Analysis (fRMA).** This method uses "frozen" RMA vectors pre-computed from great amounts of available data for the same microarray platform, accounting for those biases. However, such frozen vectors are only available for some organisms (most notably, human, mouse, zebrafish, and fluit-fly among others), but not for chicken, hence preventing the proper meta-analysis of chicken transcriptomics datasets without prior generation of own frozen parameters.

Output | Here, we present our R package - "**affyChickGenomeArrayfrmavecs**" - containing the frozen vectors that can be directly

Methods

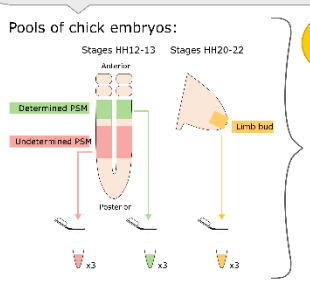


Usage



Application Example

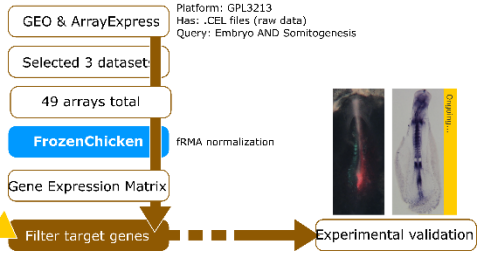
gga-miRNOME | miRNA-seq generated in our Lab



191 Differentially expressed miRNAs

3996 Different target genes

gga-transcriptome | Meta-analysis of



Conclusion

Significance| This package will directly benefit the chicken research community by facilitating future meta-analysis studies using transcriptomics datasets from public repositories, hence directly contributing to the quality of developmental research using the chick model.

Ongoing Application| This package has already been used in our lab for our current research project, showing that it is fully functional

This work was supported by FCT, Portugal (grant PTDC/BEX-BID/5410/2014) and Research Center Grant UID/BIM/04773/2013 CBMR 1334.

Annex 12 | Masters Portfolio – "ClockOME" Selected Poster

2021 | IMPSG

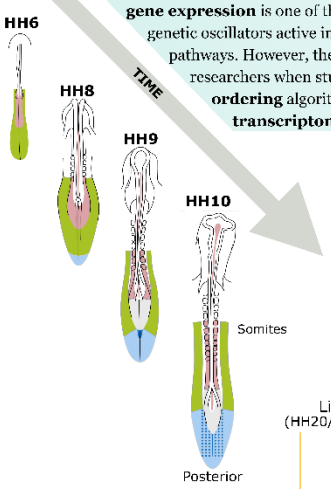
ClockOME | Searching for oscillatory genes in early vertebrate development

Marta Liber[1,2], Ramiro Magno[2,3], Raquel P. Andrade[1,2,4,5], Isabel Duarte[2,3]

1-Center for Biomedical Research (CBMR), Universidade do Algarve, Faro, Portugal
2-Algarve Biomedical Center (ABC), Universidade of Algarve, Faro, Portugal
3-Center for Health Technology and Services Research (CINTESIS), Universidade do Algarve, Faro, Portugal
4-Faculty of Medicine and Biomedical Sciences, Universidade do Algarve, Faro, Portugal
5-Algarve Biomedical Center Research Institute (ABC-RI), Faro, Portugal

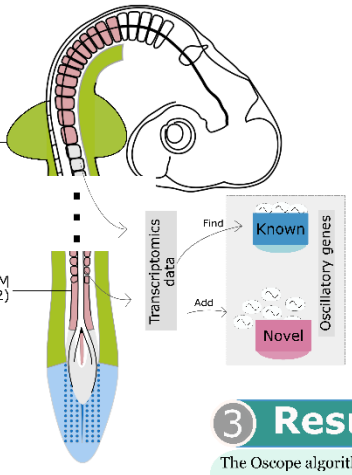
Background

Embryo development is a dynamic process regulated across **space** and **time**. Each embryo cell is constantly presented with several biochemical and mechanical signals that are translated into a cell and tissue behavior, with the ultimate objective to generate a fully functional organism. **Oscillatory gene expression** is one of the mechanisms used by the cell to **timely** regulate its behavior. The **embryo molecular clock** is one of the best known genetic oscillators active in embryo **segmentation**, involving the oscillatory/cyclic expression of genes involved in Notch, FGF, and WNT pathways. However, the list of genes with cyclic expression is still far from complete, mainly due to the experimental challenges presented to researchers when studying periodic systems for which the period is unknown. This process can be expedited by applying a **pseudotime ordering** algorithm to static transcriptomics data, i.e. conducting a **bioinformatics search** for candidate oscillatory genes using large **transcriptomics datasets** for which there is no explicit time measurement (no time-series).

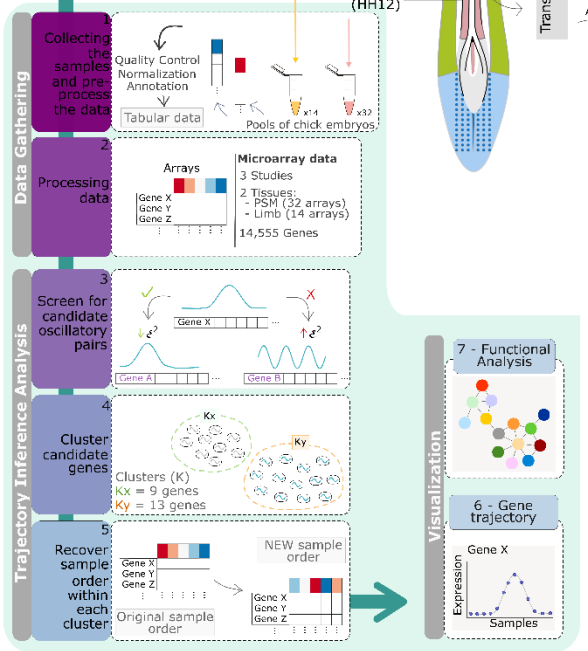


Hypothesis 1

This study aims at finding candidate oscillatory genes --collectively termed the **ClockOME** - active in early chick embryo development.



2 Approach



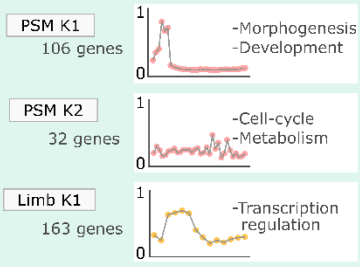
4 Significance

1) This work generated a ClockOME of 296 candidate oscillatory genes that can now be used by the research community for **guided experimental analysis** in order to discover additional members of the chick **embryo molecular clock**.

2) A short list of **six candidates** based on functional ranking has been proposed for experimental validation in the near future.

3 Results

The Oscope algorithm retrieved three clusters of potentially oscillatory genes: two clusters in PSM (comprising 126 and 32 genes) and one cluster in Limb (with 163 genes). Therefore, the ClockOME comprises a total of **296 candidate** cyclic genes. These genes belong to diverse regulatory pathways associated with morphological and developmental biological processes. The **ClockOME** presents a gene that have previously been associated with the embryo molecular clock -*Mesp2*, showing that this approach is valid; however the great majority of genes represent novel periodic candidates with functions related to chromatin and transcriptional regulation as well as with protein and oxygen metabolism.

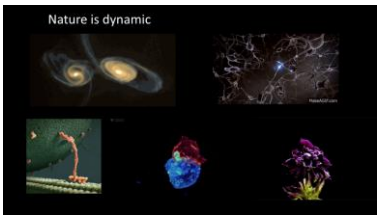


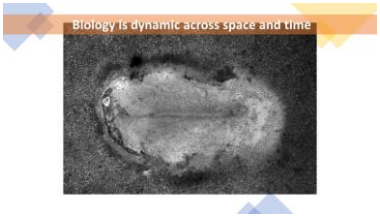
http://188.166.21.228/ClockOME_genes/

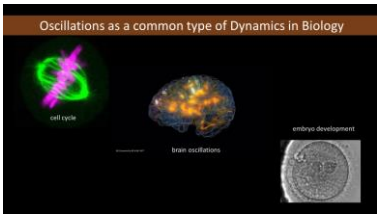
@ https://marta-liber-cv.mozello.com/
@ mlber.pt@gmail.com

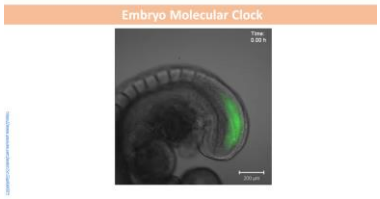
Annex 13 | Masters Portfolio – Seminars

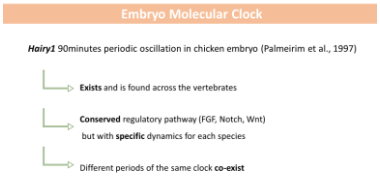
Lab Club oral presentation | March 2020

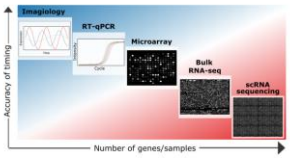


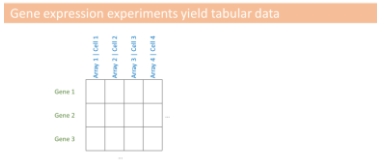


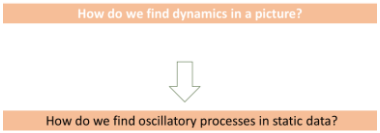


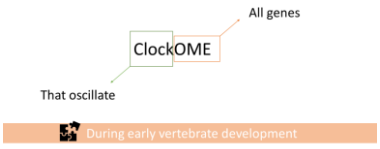


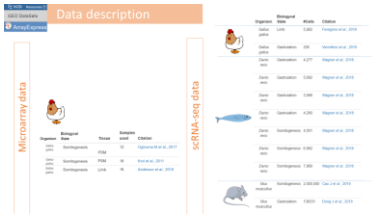


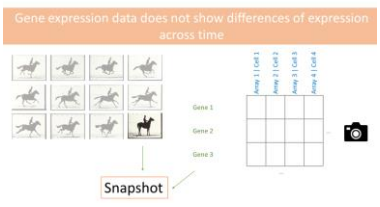


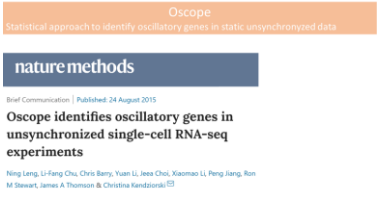


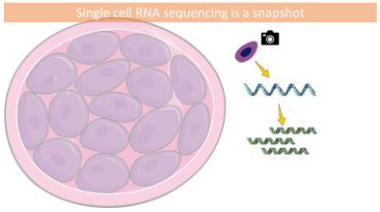


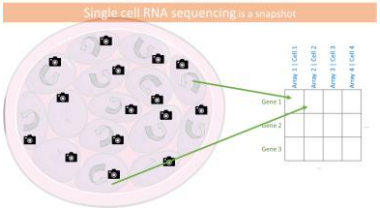


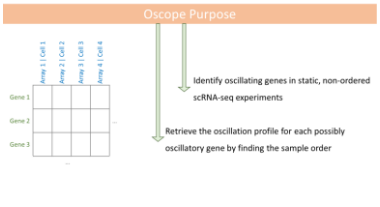




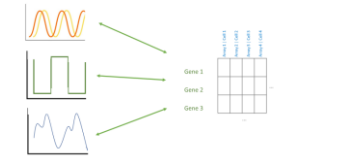






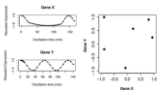


Different types of oscillations hidden in static data

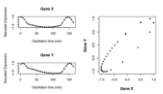


HOW

Pairwise analysis of gene expression values



Pairwise analysis of gene expression values

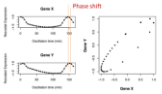


Oscope step 1

Paired-sine model

Genes that oscillate with the same frequency will form an ellipse

Pairwise analysis of gene expression values



Oscope step 1

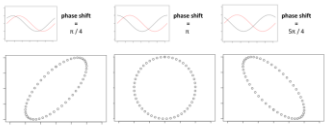
Paired-sine model

Plot 2 genes at a time in a phase plot

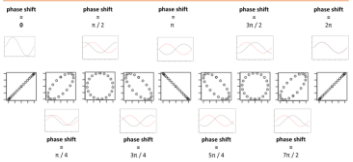
Ellipsoid form depending on the phase shift

Genes that oscillate with the same frequency will form an ellipse

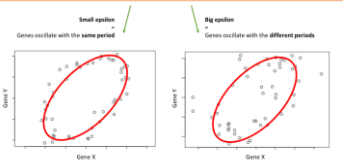
Ellipsoid form depends on the phase shift



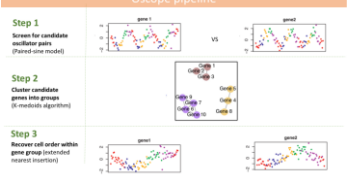
Ellipsoid form depends on the phase shift



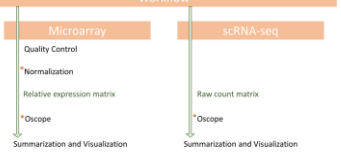
Epsilon (ε) – error term



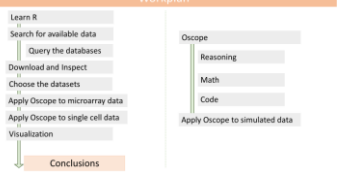
Oscope pipeline



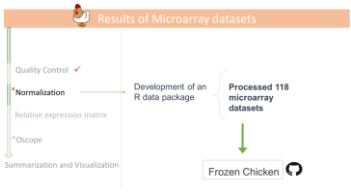
Workflow



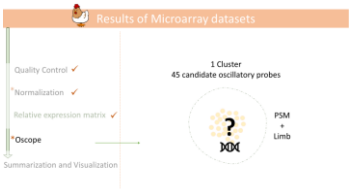
Workplan



Results of Microarray datasets

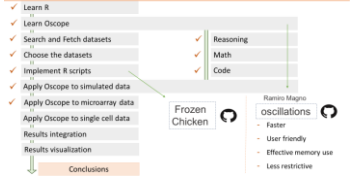


Results of Microarray datasets

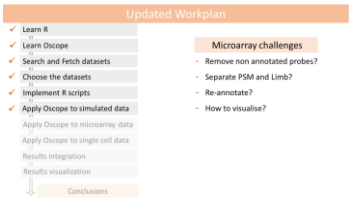


Results of scRNA-seq datasets

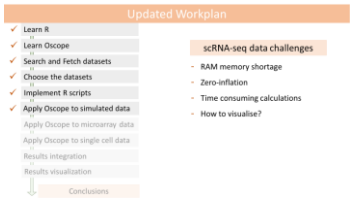
Updated Workplan



Updated Workplan



Updated Workplan



Thank you