Annex 7: Characterization

Juanma Medina

```
library("ggplot2")
library("plyr")
library("RColorBrewer")

# Read DESeq2 script until extracting the individual results
# (remember to take out the condition of at least one mapping tag)
DESeq2_res <- data.frame(res_plus, res_minus, res_control, res_kd)

# Extract only LFC
DESeq2_res <- DESeq2_res[, c(2, 8, 14, 20)]

# Remove NAs (from 11045 to 9830)
DESeq2_res <- DESeq2_res[complete.cases(DESeq2_res), ]

# Distal clusters from DE between strands correspond to higher average absolute LFC
colMeans(abs(DESeq2_res))

# Merge with cts_matrix_plus and cts_matrix_minus

# Add rownames as extra column (required by plyr)
DESeq2_res$rn <- rownames(DESeq2_res)

# Add average expression for each DHSs in control and KD conditions in both polarities
# from the 4 technical replicates, adding rownames as extra columns (required by plyr)
cts_plus_con <- data.frame(rowMeans(cts_matrix_plus[c("C0+", "C1+", "C2+", "C3+")], na.rm=TRUE))
cts_plus_con$rn <- rownames(cts_plus_con)
cts_plus_kd <- data.frame(rowMeans(cts_matrix_minus[c("C0-", "C1-", "C2-", "C3-")], na.rm=TRUE))
cts_plus_kd$rn <- rownames(cts_plus_kd)

cts_minus_con <- data.frame(rowMeans(cts_matrix_plus[c("KD0+", "KD1+", "KD2+", "KD3+")], na.rm=TRUE))
cts_minus_con$rn <- rownames(cts_minus_con)
cts_minus_kd <- data.frame(rowMeans(cts_matrix_minus[c("KD0-", "KD0-", "KD0-", "KD0-")], na.rm=TRUE))
cts_minus_kd$rn <- rownames(cts_minus_kd)

# Merging with plyr
DESeq2_res <- join_all(list(DESeq2_res, cts_plus_con, cts_plus_kd, cts_minus_con, cts_minus_kd),
                       by = "rn", type = "full")

# Remove NAs (from 11045 to 9830)
DESeq2_res <- DESeq2_res[complete.cases(DESeq2_res), ]

# rn column as rownames
full <- data.frame(DESeq2_res[, -5], row.names = DESeq2_res[, 5])

# Renaming columns
colnames(full) <- c("LFCplus", "LFCminus", "LFCcon", "LFCkd", "C+", "C-", "KD+", "KD-")

# Directionality analysis (per condition)
```

```r
full$bias_con <- full$`C+` / (full$`C+` + full$`C-`)
full$bias_kd <- full$`KD+` / (full$`KD+` + full$`KD-`)
full$dire_con <- ifelse(full$bias_con <= 0.25 | full$bias_con >= 0.75, "U", "B")
full$dire_kd <- ifelse(full$bias_kd <= 0.25 | full$bias_kd >= 0.75, "U", "B")

# Conditioning based on directionality
full$directionality <- ifelse(full$dire_con == "U" & full$dire_kd == "U", "Ud C / Ud KD",
                       ifelse(full$dire_con == "B" & full$dire_kd == "B", "Bd C / Bd KD",
                       ifelse(full$dire_con == "U" & full$dire_kd == "B",
                                           "Ud C / Bd KD", "Bd C / Ud KD")))

# Remove 36 NAs (9794)
full <- full[complete.cases(full), ]

# Number of uni and bidirectionally transcribed DHSs in each condition
# 39.3% and 45.6% BDTrans
sum(full$dire_con == "U"); sum(full$dire_kd == "U")
sum(full$dire_con == "B"); sum(full$dire_kd == "B")

# Exosome sensitivity (per polarity)
full$type_plus <- (full$`KD+` - full$`C+`) / full$`KD+`
full$type_minus <- (full$`KD-` - full$`C-`) / full$`KD-`

full$exsens_plus <- ifelse(full$type_plus <= 0.25, "Stb",
                           ifelse(full$type_plus >= 0.75, "Uns", "Unc"))
full$exsens_minus <- ifelse(full$type_minus <= 0.25, "Stb",
                           ifelse(full$type_minus >= 0.75, "Uns", "Unc"))

# Conditioning based on stability
full$stability <- ifelse(full$exsens_plus == "Stb" & full$exsens_minus == "Stb", "Stb + / Stb -",
                  ifelse(full$exsens_plus == "Stb" & full$exsens_minus == "Uns", "Stb + / Uns -",
                  ifelse(full$exsens_plus == "Uns" & full$exsens_minus == "Stb", "Uns + / Stb -",
                  ifelse(full$exsens_plus == "Uns" & full$exsens_minus == "Uns", "Uns + / Uns -",
                                          "NonC"))))

# Remove 256 NAs (9538)
full <- full[complete.cases(full), ]

# Number of sensitive and exosome unsensitive transcribed DHSs in each condition
sum(full$exsens_plus == "Stb"); sum(full$exsens_plus == "Uns"); sum(full$exsens_plus == "Unc")
sum(full$exsens_minus == "Stb"); sum(full$exsens_minus == "Uns"); sum(full$exsens_minus == "Unc")

# PCA
pca_res <- prcomp(x = full[, 1:4], scale = T, center = T)
summary(pca_res)$importance[2, ]
prop_var <- summary(pca_res)$importance[2, ]

theme_set(theme_bw())

# PCA coloring according the directionality
qplot(data = data.frame(pca_res$x), x = PC1, y = PC2, color = full$directionality) +
  labs(x = paste("PC1", round(prop_var[1] * 100, digits = 2), "%", sep = " "),
       y = paste("PC2", round(prop_var[2] * 100, digits = 2), "%", sep = " ")) +
```

```
  xlim(-3, 3) + ylim(-2.5, 2.5) + labs(colour = 'Directionality') +
  ggtitle("PCA of DHSs by directionality") +
  theme_bw(base_size = 8) +
  theme(legend.justification = "right",
        legend.margin = margin(-6, -6, -6, -6),
        legend.box.margin = margin(3, 3, 3, 3))

# PCA coloring according the exosome sensitivity
qplot(data = data.frame(pca_res$x), x = PC1, y = PC2, color = full$stability) +
  labs(x = paste("PC1", round(prop_var[1] * 100, digits = 2), "%", sep = " "),
       y = paste("PC2", round(prop_var[2] * 100, digits = 2), "%", sep = " ")) +
  xlim(-3, 3) + ylim(-2.5, 2.5) + labs(colour = 'Stability') +
  ggtitle("PCA of DHSs by exosome sensitivity") +
  theme_bw(base_size = 8) +
  theme(legend.justification = "right",
        legend.margin = margin(-6, -6, -6, -6),
        legend.box.margin = margin(3, 3, 3, 3))

# Clustering according to LFC experiments
set.seed(20)
my_cluster <- kmeans(full[, 1:4], 4, nstart = 20)
my_cluster$cluster <- as.factor(my_cluster$cluster)

# LFC clustering applied here
qplot(data = data.frame(pca_res$x), x = PC1, y = PC2, color = my_cluster$cluster) +
  labs(x = paste("PC1", round(prop_var[1] * 100, digits = 2), "%", sep = " "),
       y = paste("PC2", round(prop_var[2] * 100, digits = 2), "%", sep = " ")) +
  xlim(-3, 3) + ylim(-2.5, 2.5) + labs(colour = 'LFC cluster') +
  ggtitle("PCA of DHSs by LFC clustering") +
  theme_bw(base_size = 8) +
  theme(legend.justification = "right",
        legend.margin = margin(-6, -6, -6, -6),
        legend.box.margin = margin(3, 3, 3, 3))

# CHANGE OF PARAMETERS HERE
# Make a simpler version of directionality
full$directionality <- ifelse(full$dire_con == "U" & full$dire_kd == "U", "Unidirectional",
                       ifelse(full$dire_con == "B" & full$dire_kd == "B", "Bidirectional",
                                    "KD dependant"))

# Change TH of exosome sensitivity limits
full$exsens_plus <- ifelse(full$type_plus <= 0.5, "Stb", "Uns")
full$exsens_minus <- ifelse(full$type_minus <= 0.5, "Stb", "Uns")

# Make a simpler version of sensitivity classification
full$stability <- ifelse(full$exsens_plus == "Uns" & full$exsens_minus == "Uns", "Weak",
                  ifelse(full$exsens_plus == "Stb" & full$exsens_minus == "Stb", "Strong",
                               "Average"))

# Wombo combo
full$final <- ifelse(full$directionality == "Bidirectional" & full$stability == "Strong", "BD / Stable"
              ifelse(full$directionality == "Bidirectional" & full$stability == "Weak", "BD / Weak",
              ifelse(full$directionality == "Unidirectional" & full$stability == "Strong", "UD / Stable
```

```
            ifelse(full$directionality == "Unidirectional" & full$stability == "Weak", "UD / Weak",
            ifelse(full$directionality == "KD dependant" & full$stability == "Strong", "KD_dep / Stabl
            ifelse(full$directionality == "KD dependant" & full$stability == "Weak", "KD_dep / Weak",
            ifelse(full$directionality == "Bidirectional" & full$stability == "Average", "BD / Averag
            ifelse(full$directionality == "Unidirectional" & full$stability == "Average", "UD / Averag
                  "KD_dep / Average")))))))

# PCA coloring according directionality and exosome sensitivity
qplot(data = data.frame(pca_res$x), x = PC1, y = PC2, color = full$final) +
  scale_color_brewer(palette = "Set1") +
  labs(x = paste("PC1", round(prop_var[1] * 100, digits = 2), "%", sep = " "),
       y = paste("PC2", round(prop_var[2] * 100, digits = 2), "%", sep = " ")) +
  xlim(-3, 3) + ylim(-2.5, 2.5) + labs(colour = 'DHS cluster') +
  ggtitle("PCA of DHSs fully clustered") +
  theme_bw(base_size= 8) +
  theme(legend.justification = "right",
        legend.margin = margin(-6, -6, -6, -6),
        legend.box.margin=margin(3, 3, 3, 3))
```