EMTAB3929: Aneuploidy Analysis

Margaret R. Starostik

Data Check

Double-check the processed EMTAB3929 data (ProcessedData/EMTAB3929_DataPrep.RData) before moving on with the aneuploidy test.
(1) Annotation: 56400 genes
(2) Gene expression matrix in counts: 2991 genes and 1481 cells
(3) Gene expression matrix in CPM: 2991 genes and 1481 cells
(4) Gene expression matrix in $\log_2$(CPM+1): 2991 genes and 1481 cells
(5) Metasheet: 1481 cells and 88 embryos

Aneuploidy Test

Ploidy analysis was performed using the scploid R package. Data substructure was taken into account to avoid confounding the ploidy analysis by unrelated transcriptional differences, and so EMTAB3929 cells were analyzed in groups according to embryonic stage and cell lineage.

```
test_table <- table(metasheet$EStage, metasheet$`Revised lineage (this study)`)
knitr::kable(test_table, caption = "Aneuploidy Tests", row.names = TRUE)
```

Table 1: Aneuploidy Tests

	Epiblast	ICM	Intermediate	Primitive Endoderm	Trophectoderm	Undefined
E3	0	0	0	0	0	78
E4	0	0	0	0	0	185
E5	0	66	0	0	227	67
E6	25	0	36	8	336	0
E7	20	0	32	22	379	0

```
# Make the ploidytest object. Note that the gene expression
# matrix provided is based on counts that have not been
# filtered based on expression threshold. Check that all
# information among various data sets are correctly
# organized.
emtab3929_counts <- emtab3929_counts[, colnames(emtab3929_counts) %in%
    metasheet$Sample]
emtab3929_counts <- emtab3929_counts[order(rownames(emtab3929_counts)),
    order(colnames(emtab3929_counts))]
annotation <- annotation[order(annotation$ensembl_gene_id), ]
metasheet <- metasheet[order(metasheet$Sample), ]
metasheet$Group <- paste0(metasheet$EStage, "_", metasheet$`Revised lineage (this study)`)  # to split data

ploidytest <- makeAneu(counts = emtab3929_counts, genes = annotation$ensembl_gene_id,
    chrs = annotation$chromosome_name, cellNames = metasheet$Sample,
    cellGroups = metasheet$Group)  # split data by EStage and tissue type
```

```
# Go through `doAneu` function step-by-step to understand how
# the data are split and analyzed
spt <- splitCellsByGroup(ploidytest)  # data split into 13 subsets, one for each EStage_tissue combination
results <- do.call(rbind, lapply(spt, calcAneu))
results$p.adj <- p.adjust(results$p, method = "fdr")
results$monosomy = results$z < 0
hits = results[results$p.adj < getParam(ploidytest, "p.thresh") &
    abs(results$score - 1) > getParam(ploidytest, "min.deviation"),
    ]
ploidytest@scores = results
ploidytest@aneuploidies = hits

# Put together results.  (1) Hits only.
aneuploidy_hits <- getHits(ploidytest)  # 1,705 hits
output_hits <- inner_join(metasheet, aneuploidy_hits, by = c(Sample = "cell"))
colnames(output_hits)[colnames(output_hits) == "chr"] <- "AneuploidyTest_chromosome"
colnames(output_hits)[colnames(output_hits) == "z"] <- "AneuploidyTest_zscore"
colnames(output_hits)[colnames(output_hits) == "score"] <- "AneuploidyTest_score"
colnames(output_hits)[colnames(output_hits) == "p"] <- "AneuploidyTest_pvalue"
colnames(output_hits)[colnames(output_hits) == "p.adj"] <- "AneuploidyTest_FDR"
colnames(output_hits)[colnames(output_hits) == "monosomy"] <- "AneuploidyTest_monosomy"
write.table(output_hits, file = paste0(project_folder, "Results/03_AneuploidyTest/AneuploidyTest_HitsOnly.txt"),
    sep = "\t", quote = FALSE, row.names = FALSE)

# (2) All scores.  table(output_results$EStage,
# output_results$AneuploidyTest_chromosome). E3=78, E4=167,
# E5=360, E6=405, E7=453 --> 32186 results predicted.
aneuploidy_results <- as.data.frame(getScores(ploidytest))
output_results <- inner_join(metasheet, aneuploidy_results, by = c(Sample = "cell"))
colnames(output_results)[colnames(output_results) == "chr"] <- "AneuploidyTest_chromosome"
colnames(output_results)[colnames(output_results) == "z"] <- "AneuploidyTest_zscore"
colnames(output_results)[colnames(output_results) == "score"] <- "AneuploidyTest_score"
colnames(output_results)[colnames(output_results) == "p"] <- "AneuploidyTest_pvalue"
colnames(output_results)[colnames(output_results) == "p.adj"] <- "AneuploidyTest_FDR"
colnames(output_results)[colnames(output_results) == "monosomy"] <- "AneuploidyTest_monosomy"

# Modify output so that the actual aneuploidy hits are easily
# distinguished from rest of results.
no_hits <- dplyr::setdiff(output_results, output_hits)
dim(no_hits)  # 30,877
no_hits$AneuploidyTest_hit <- "no"

hits <- output_hits
hits$AneuploidyTest_hit <- "yes"
dim(hits)  # 1,705

# Sanity check.
dim(no_hits)[1] + dim(hits)[1] == dim(output_results)[1]  # TRUE. Good.

# Compile hits and no hits results.
modified_output <- rbind(hits, no_hits)
dim(modified_output)  # 32,582
modified_output$Ploidy <- 2  # disomy
```

```
modified_output$Ploidy[modified_output$AneuploidyTest_FDR < 0.05 &
    modified_output$AneuploidyTest_monosomy == TRUE] <- 1  # monosomy
modified_output$Ploidy[modified_output$AneuploidyTest_FDR < 0.05 &
    modified_output$AneuploidyTest_monosomy == FALSE] <- 3  # trisomy

table(modified_output$AneuploidyTest_hit, modified_output$Ploidy)

write.table(modified_output, file = paste0(project_folder, "Results/03_AneuploidyTest/AneuploidyTest_AllScores.txt"),
    sep = "\t", quote = FALSE, row.names = FALSE)

# (3) Aneuploidy test metrics.
write.table(getMetrics(ploidytest), file = paste0(project_folder,
    "Results/03_AneuploidyTest/AneuploidyTest_Metrics.txt"),
    sep = "\t", quote = FALSE)

write.table(data.frame(row.names = names(assessMetrics(ploidytest)),
    result = assessMetrics(ploidytest)), file = paste0(project_folder,
    "Results/03_AneuploidyTest/AneuploidyTest_MetricsSummary.txt"),
    sep = "\t", quote = FALSE)
```

Heatmaps

Generate clustered heatmaps for each embryo. Specifically, depict heatmap of aneuploidy results for each embryo, clustered by cells across all chromosomes. Chromosomes are thought to be independent, so do not cluster by chromosomes, just by cells.

```
embryos <- unique(modified_output$Embryo) # 88

for (i in 1:length(embryos)){
  sdata <- modified_output[modified_output$Embryo == embryos[i], ]
  sdata <- data.frame(sdata)

  heatmap_matrix <- sdata[, c(1, 14, 21)] %>%
    spread(AneuploidyTest_chromosome, Ploidy)
  rownames(heatmap_matrix) <- heatmap_matrix$Sample
  heatmap_matrix <- heatmap_matrix[, -1]
  heatmap_matrix <- data.matrix(heatmap_matrix)


  embryo_folder <- sapply(strsplit(embryos, "_"), "[", 1)

  # can only generate heatmaps if there is more than 1 result
  if (dim(heatmap_matrix)[1] >= 2){
     # Colors if monosomy, disomy, and trisomy present
    if(length(unique(c(heatmap_matrix))) == 3){
      my_colors <- c(rgb(82, 125, 157, maxColorValue = 255),
                     rgb(224, 224, 224, maxColorValue = 255),
                     rgb(183, 94, 81, maxColorValue = 255))
      key_label <- c("M", "D", "T")
      at_label <- c(0.15, 0.5, 0.85)
    }
    # Colors if only two of the three ploidies are present
    if (length(unique(c(heatmap_matrix))) == 2){
```

```
  # monosomy and disomy present
  if (is.element(1, unique(c(heatmap_matrix))) & is.element(2, unique(c(heatmap_matrix)))) {
    my_colors <- c(rgb(82, 125, 157, maxColorValue = 255), rgb(224, 224, 224, maxColorValue = 255))
    key_label <- c("M", "D")
    at_label <- c(0.25, 0.75)
  }
  # monosomy and trisomy present
  else if (is.element(1, unique(c(heatmap_matrix))) & is.element(3, unique(c(heatmap_matrix)))){
    my_colors <- c(rgb(82, 125, 157, maxColorValue = 255), rgb(183, 94, 81, maxColorValue = 255))
    key_label <- c("M", "T")
    at_label <- c(0.25, 0.75)
  }
  # disomy and trisomy present
  else if (is.element(2, unique(c(heatmap_matrix))) & is.element(3, unique(c(heatmap_matrix)))){
    my_colors <- c(rgb(224, 224, 224, maxColorValue = 255), rgb(183, 94, 81, maxColorValue = 255))
    key_label <- c("D", "T")
    at_label <- c(0.25, 0.75)
  }
}
# Colors only if one of the two ploidies are present
if(length(unique(c(heatmap_matrix))) == 1){
  # only monosomy present
  if (is.element(1, unique(c(heatmap_matrix)))) {
    my_colors <- rep(rgb(82, 125, 157, maxColorValue = 255), 2)
    key_label <- "M"
    at_label <- 0.50
  }
  # only disomy present
  else if (is.element(2, unique(c(heatmap_matrix)))) {
    my_colors <- rep(rgb(224, 224, 224, maxColorValue = 255), 2)
    key_label <- "D"
    at_label <- 0.50
  }
  # only trisomy present
  else if (is.element(3, unique(c(heatmap_matrix)))) {
    my_colors <- rep(rgb(183, 94, 81, maxColorValue = 255), 2)
    key_label <- "T"
    at_label <- 0.50
  }
}

distance.row <- dist(heatmap_matrix, method = "euclidean") # same parameters as honeyBADGER
cluster.row <- hclust(distance.row, method = "ward.D") # same parameters as honeyBADGER

sample_order <- rev(cluster.row$labels[cluster.row$order])
cell_lineage <- metasheet[, c(1, 9)][match(sample_order, metasheet$Sample), ]
cell_lineage$colors <- ifelse(cell_lineage$`Revised lineage (this study)` == "Undetermined", "#B3E2CD",
                                ifelse(cell_lineage$`Revised lineage (this study)` == "ICM", "#FDCDAC",
                                       ifelse(cell_lineage$`Revised lineage (this study)` == "Trophectoderm", "#CBD5E8",
                                              ifelse(cell_lineage$`Revised lineage (this study)` == "Intermediate", "#F4CAE4",
                                                     ifelse(cell_lineage$`Revised lineage (this study)` == "Epiblast", "#E6F5C9", "#FFF2AE")))))

pdf(paste0(project_folder, "Results/03_AneuploidyTest/",
        embryo_folder[i], "/CalledAneuploidChromosomes_ClusteredHeatmap_", embryos[i], ".pdf"),
    height = 10, width = 10)
  heatmap.2(heatmap_matrix,

       # dendrogram control
       dendrogram = "row",
       Rowv = as.dendrogram(cluster.row),
       Colv = FALSE,

       # level trace
       trace = "none",

       # data scaling
       scale = "none",

       # color key and density info
       keysize = 0.1,
       key.xlab = "",
       key.xtickfun = function(){
         return(list(labels = key_label, at = at_label, tick = FALSE))
       },
       density.info = "none",

       # plot layout
       lhei = c(0.5, 1),
       lwid = c(0.6, 1),

       # colors
       col = my_colors,

       # plot labels and set sizes
       xlab = "Chromosome",
       margins = c(8, 12),

       # row/column labeling ()
       RowSideColors = cell_lineage$colors,
       srtCol = 0,
       adjCol = c(0.5, 0.5))
  legend("topright", legend = unique(cell_lineage$`Revised lineage (this study)`),
      col = unique(cell_lineage$colors), lty = 1, lwd = 0.5, cex = 0.7)
  dev.off()
 }
}
```