Cardiomyocyte interactome, additional KEGG & Gene Ontology analyses

Sebastian Kurscheid

1 July 2015

R script for KEGG pathway analysis of cardiomyocyte RNA interactome proteins

Loading of the data tables.

```
load("data/kegg.brite.rda")
load("data/interactome.rda")
load("data/wcl.rda")
```

Fetching KEGG identifiers:

```
ids <- unlist(lapply(strsplit(kegg.brite$C, " "), function(x) x[1]))
rownames(kegg.brite) <- ids
total.keggIDs <- keggLink("mmu", "pathway")
save(total.keggIDs, file = "data/total.keggIDs.rda")
```

We have found a total of `length(total.keggIDs)` which are used for mapping the WCL and interactome data.

Now we proceed to mapping of WCL protein IDs to KEGG IDs and testing for enrichments against background of all KEGG proteins contained in KEGG pathways.

```
# retrieved Entrez IDs from Biomart
mouse <- useMart("ensembl", dataset = "mmusculus_gene_ensembl")
human <- useMart("ensembl", dataset = "hsapiens_gene_ensembl")
attribs <- listAttributes(mouse)
pages <- attributePages(mouse)
hsap.attribs <- listAttributes(human)

entrez_ids <- getBM(attributes = c("ensembl_gene_id", "entrezgene"), values = wcl[,
    "ensembl_gene_id"], filters = "ensembl_gene_id", mart = mouse)
wcl.human_homologs <- getBM(attributes = c("ensembl_gene_id", "hsapiens_homolog_ensembl_gene"),
    values = wcl[, "ensembl_gene_id"], filters = "ensembl_gene_id", mart = mouse)

# remove ensembl_gene_ids which have duplicated entrez_ids
entrez_ids <- entrez_ids[-which(duplicated(entrez_ids$ensembl_gene_id)), ]
wcl <- merge(wcl, entrez_ids, by.x = "ensembl_gene_id", by.y = "ensembl_gene_id",
    all.x = T)
wcl.entrezIDs <- unique(wcl[!is.na(wcl$entrezgene), ]$entrezgene)
# this retrieval is fairly slow, therefore the results were written to
# './data'
wcl.keggIDs <- keggConv.batch(wcl.entrezIDs)
save(wcl.keggIDs, file = "data/wcl.keggIDs.rda")
```

```
wcl.keggQ <- lapply(wcl.keggIDs, function(x) keggGet(x))
save(wcl.keggQ, file = "data/wcl.keggQ.rda")

wcl.pathways <- unique(unlist(lapply(strsplit(names(unlist(lapply(wcl.keggQ,
    function(x) x[[1]]$PATHWAY))), "\\."), function(x) x[3])))
save(wcl.pathways, file = "data/wcl.pathways.rda")

wcl.pathways.genes <- lapply(wcl.pathways, function(x) keggLink("genes", x))
names(wcl.pathways.genes) <- wcl.pathways
save(wcl.pathways.genes, file = "data/wcl.pathways.genes.rda")

wcl.pathways.genes.entrez_ids <- unique(gsub("mmu:", "", as.character(unlist(wcl.pathways.genes))))
wcl.df <- kegg.brite[gsub("mmu", "", wcl.pathways), ]
wcl.df$ID <- rownames(wcl.df)
wcl.df$total <- rep(0, nrow(wcl.df))
wcl.df$total <- sapply(rownames(wcl.df), function(x) length(wcl.pathways.genes[[paste("mmu",
    x, sep = "")]]))
wcl.df$count <- rep(0, nrow(wcl.df))
wcl.df$frac <- rep(0, nrow(wcl.df))

for (i in rownames(wcl.df)) {
    # print(i)
    kL1 <- keggLink("mmu", paste("mmu", i, sep = ""))
    wcl.df[i, ]$count <- length(which(wcl.keggIDs %in% kL1))
    wcl.df[i, ]$frac <- round(length(which(wcl.keggIDs %in% kL1))/length(kL1) *
        100, 2)
}

# extract list of IDs in pathway
wcl.in_path.IDs <- lapply(rownames(wcl.df), function(x) {
    kL1 <- keggLink("mmu", paste("mmu", x, sep = ""))
    in_path <- wcl.keggIDs[which(wcl.keggIDs %in% kL1)]
})

names(wcl.in_path.IDs) <- rownames(wcl.df)

# perform Fisher's Exact Test for each category
bkgd <- length(unique(total.keggIDs))
smpl <- length(wcl.keggIDs)
ftl <- apply(wcl.df[1, ], 1, function(x) {
    ct <- as.integer(x["count"])
    tt <- as.integer(x["total"])
    m1 <- matrix(c(ct, tt, smpl - ct, bkgd - tt), 2, 2)
    fisher.test(m1, alternative = alternative)
})

wcl.df$ft_pval <- unlist(lapply(ftl, function(x) {
    x$p.value
}))
wcl.df$ft_OR <- unlist(lapply(ftl, function(x) {
    x$estimate
}))
wcl.df$ft_fdr <- p.adjust(wcl.df$ft_pval, method = "fdr")
```

```
save(wcl.df, file = "data/wcl.df.rda")
```

Mapping of Interactome protein IDs to KEGG IDs and testing for enrichments against background of WCL proteins contained in KEGG pathways.

```
interactome.entrez_ids <- getBM(attributes = c("ensembl_gene_id", "entrezgene"),
    values = interactome[, "ensembl_gene_id"], filters = "ensembl_gene_id",
    mart = mouse)
interactome.human_homologs <- getBM(attributes = c("ensembl_gene_id", "hsapiens_homolog_ensembl_gene"),
    values = interactome[, "ensembl_gene_id"], filters = "ensembl_gene_id",
    mart = mouse)

# remove ensembl_gene_ids which have duplicated entrez_ids
interactome.entrez_ids <- interactome.entrez_ids[-which(duplicated(interactome.entrez_ids$ensembl_gene_
    ]
interactome <- merge(interactome, interactome.entrez_ids, by.x = "ensembl_gene_id",
    by.y = "ensembl_gene_id", all.x = T)

interactome.entrezIDs <- unique(interactome[!is.na(interactome$entrezgene),
    ]$entrezgene)
save(interactome.entrezIDs, file = "data/interactome.entrezIDs")

interactome.keggIDs <- keggConv.batch(interactome.entrezIDs)
save(interactome.keggIDs, file = "data/interactome.keggIDs.rda")

interactome.keggQ <- lapply(interactome.keggIDs, function(x) keggGet(x))
save(interactome.keggQ, file = "data/interactome.keggQ.rda")

interactome.pathways <- unique(unlist(lapply(strsplit(names(unlist(lapply(interactome.keggQ,
    function(x) x[[1]]$PATHWAY))), "\\."), function(x) x[3])))
save(interactome.pathways, file = "data/interactome.pathways.rda")

interactome.pathways.genes <- lapply(interactome.pathways, function(x) keggLink("genes",
    x))
names(interactome.pathways.genes) <- interactome.pathways
save(interactome.pathways.genes, file = "data/interactome.pathways.genes.rda")

interactome.pathways.genes.entrez_ids <- unique(gsub("mmu:", "", as.character(unlist(interactome.pathway


# create dataframe for counting hits in pathways
interactome.df <- kegg.brite[gsub("mmu", "", interactome.pathways), ]
interactome.df$source <- rep("Interactome", nrow(interactome.df))
interactome.df$ID <- rownames(interactome.df)

# we are now using WCL as background to test for enrichment
i1 <- intersect(rownames(interactome.df), rownames(wcl.df))
interactome.df$total <- rep(0, nrow(interactome.df))
interactome.df[i1, ]$total <- wcl.df[i1, ]$count
interactome.df$count <- rep(0, nrow(interactome.df))
interactome.df$frac <- rep(0, nrow(interactome.df))

for (i in rownames(interactome.df)) {
```

```
    kL1 <- keggLink("mmu", paste("mmu", i, sep = ""))
    interactome.df[i, ]$count <- length(which(interactome.keggIDs %in% kL1))
    interactome.df[i, ]$frac <- round(length(which(interactome.keggIDs %in%
        kL1))/length(kL1) * 100, 2)
}

# extract list of IDs in pathway
interactome.in_path.IDs <- lapply(rownames(interactome.df), function(x) {
    kL1 <- keggLink("mmu", paste("mmu", x, sep = ""))
    in_path <- interactome.keggIDs[which(interactome.keggIDs %in% kL1)]
})

# perform Fisher's Exact Test for each category
bkgd <- length(unique(wcl.keggIDs))
smpl <- length(interactome.keggIDs)

ftl <- apply(interactome.df, 1, function(x) {
    ct <- as.integer(x["count"])
    tt <- as.integer(x["total"])
    m1 <- matrix(c(ct, tt, smpl - ct, bkgd - tt), 2, 2)
    fisher.test(m1, alternative = alternative)
})

interactome.df$ft_pval <- unlist(lapply(ftl, function(x) {
    x$p.value
}))
interactome.df$ft_OR <- unlist(lapply(ftl, function(x) {
    x$estimate
}))
interactome.df$ft_fdr <- p.adjust(interactome.df$ft_pval, method = p.adjust.method,
    n = nrow(wcl.df))
save(interactome.df, file = "data/interactome.df.rda")
```

KEGG contains three levels/hierarchies (A>B>C), here we summarize the enrichment at B level:

```
# ---------interactome-summarizing data at 'B' level before doing Fisher's
# Exact test------
interactome.B.df <- data.frame(matrix(ncol = 5, nrow = length(unique(interactome.df$B))))
colnames(interactome.B.df) <- c("B", "A", "total", "count", "source")
interactome.B.df$B <- unique(interactome.df$B)
interactome.B.df$A <- sapply(unique(interactome.df$B), function(x) {
    A <- unique(interactome.df[which(interactome.df$B %in% x), "A"])
})
interactome.B.df$source <- rep("Interactome", nrow(interactome.B.df))
interactome.B.df$total <- sapply(unique(interactome.df$B), function(x) {
    tot <- sum(interactome.df[which(interactome.df$B %in% x), "total"])
})
interactome.B.df$count <- sapply(unique(interactome.df$B), function(x) {
    count <- sum(interactome.df[which(interactome.df$B %in% x), "count"])
})

bkgd <- length(unique(wcl.keggIDs))
smpl <- length(interactome.keggIDs)
```

```
ftl <- apply(interactome.B.df, 1, function(x) {
    ct <- as.integer(x["count"])
    tt <- as.integer(x["total"])
    m1 <- matrix(c(ct, tt, smpl - ct, bkgd - tt), 2, 2)
    fisher.test(m1, alternative = alternative)
})

interactome.B.df$ft_pval <- unlist(lapply(ftl, function(x) {
    x$p.value
}))
interactome.B.df$ft_OR <- unlist(lapply(ftl, function(x) {
    x$estimate
}))
interactome.B.df$ft_fdr <- p.adjust(interactome.B.df$ft_pval, method = p.adjust.method,
    n = nrow(wcl.df))
save(interactome.B.df, file = "data/interactome.B.df.rda")
```

We are splitting the Interactome proteins into RNA-related and un-related proteins, based on annotation analysis (upstream of these steps): First, RNA-unrelated

```
#-----------GO RNA unrelated-----------------------------------------------
# subset the interactome table
interactome.go_rna_unrelated <- interactome[which(interactome$GO == "unrelated"),]
interactome.go_rna_unrelated.entrezIDs <- unique(interactome.go_rna_unrelated[!is.na(interactome.go_rna
interactome.go_rna_unrelated.keggIDs <- keggConv.batch(interactome.go_rna_unrelated.entrezIDs)

# dataframe for count data
interactome.go_rna_unrelated.df <- interactome.df
interactome.go_rna_unrelated.df$source <- rep("GO_RNA_unrelated", nrow(interactome.go_rna_unrelated.df))
interactome.go_rna_unrelated.df$ID <- rownames(interactome.go_rna_unrelated.df)
interactome.go_rna_unrelated.df$total <- rep(0, nrow(interactome.go_rna_unrelated.df))

# we are now using WCL as background to test for enrichment
i1 <- intersect(rownames(interactome.go_rna_unrelated.df), rownames(wcl.df))
interactome.go_rna_unrelated.df[i1,]$total <- wcl.df[i1,]$count
interactome.go_rna_unrelated.df$count <- rep(0, nrow(interactome.go_rna_unrelated.df))

for (i in rownames(interactome.go_rna_unrelated.df)) {
  kL1 <- keggLink("mmu", paste("mmu", i, sep = ""))
  interactome.go_rna_unrelated.df[i, ]$count <- length(which(interactome.go_rna_unrelated.keggIDs %in% k
}

# extract list of IDs in pathway
interactome.go_rna_unrelated.in_path.IDs <- lapply(rownames(interactome.go_rna_unrelated.df), function(i
  kL1 <- keggLink("mmu", paste("mmu", x, sep = ""))
  in_path <- interactome.go_rna_unrelated.keggIDs[which(interactome.go_rna_unrelated.keggIDs %in% kL1)]
})
names(interactome.go_rna_unrelated.in_path.IDs) <- rownames(interactome.go_rna_unrelated.df)

# perform Fisher's Exact Test for each category
# Using WCL as background
bkgd <- length(unique(wcl.keggIDs))
smpl <- length(interactome.go_rna_unrelated.keggIDs)
```

```
ftl <- apply(interactome.go_rna_unrelated.df, 1, function (x) {
  ct <- as.integer(x["count"])
  tt <- as.integer(x["total"])
  m1 <- matrix(c(ct, tt, smpl - ct, bkgd - tt), 2, 2)
  fisher.test(m1, alternative = alternative)
})

interactome.go_rna_unrelated.df$ft_pval <- unlist(lapply(ftl, function(x) {x$p.value}))
interactome.go_rna_unrelated.df$ft_OR <- unlist(lapply(ftl, function(x) {x$estimate}))
interactome.go_rna_unrelated.df$ft_fdr <- p.adjust(interactome.go_rna_unrelated.df$ft_pval, method = p.a
save(interactome.go_rna_unrelated.df, file = "data/interactome.go_rna_unrelated.df.rda")

# summarizing data at "B" level before doing Fisher's Exact test
interactome.go_rna_unrelated.B.df <- data.frame(matrix(ncol = 5, nrow = length(unique(interactome.go_rna
colnames(interactome.go_rna_unrelated.B.df) <- c("B", "A", "total", "count", "source")
interactome.go_rna_unrelated.B.df$B <- unique(interactome.go_rna_unrelated.df$B)
interactome.go_rna_unrelated.B.df$A <- sapply(unique(interactome.go_rna_unrelated.df$B), function(x) {A
interactome.go_rna_unrelated.B.df$source <- rep("GO_RNA_unrelated", nrow(interactome.go_rna_unrelated.B
interactome.go_rna_unrelated.B.df$total <- sapply(unique(interactome.go_rna_unrelated.df$B), function(x
interactome.go_rna_unrelated.B.df$count <- sapply(unique(interactome.go_rna_unrelated.df$B), function(x

# using WCL as background
ftl <- apply(interactome.go_rna_unrelated.B.df, 1, function (x) {
  ct <- as.integer(x["count"])
  tt <- as.integer(x["total"])
  m1 <- matrix(c(ct, tt, smpl - ct, bkgd - tt), 2, 2)
  fisher.test(m1, alternative = alternative)
})

interactome.go_rna_unrelated.B.df$ft_pval <- unlist(lapply(ftl, function(x) {x$p.value}))
interactome.go_rna_unrelated.B.df$ft_OR <- unlist(lapply(ftl, function(x) {x$estimate}))
interactome.go_rna_unrelated.B.df$ft_fdr <- p.adjust(interactome.go_rna_unrelated.B.df$ft_pval, method =
save(interactome.go_rna_unrelated.B.df, file = "data/interactome.go_rna_unrelated.B.df")
```

Now, RNA-related:

```
#----------GO RNA related-------------------------------------------
interactome.go_rna_related <- interactome[-which(interactome$GO == "unrelated"),]

interactome.go_rna_related.entrezIDs <- unique(interactome.go_rna_related[!is.na(interactome.go_rna_rela
interactome.go_rna_related.keggIDs <- keggConv.batch(interactome.go_rna_related.entrezIDs)

# we are testing this subset of "interactome", therefore we include all the pathways from "interactome"
interactome.go_rna_related.df <- interactome.df
i1 <- intersect(rownames(interactome.go_rna_related.df), rownames(wcl.df))
interactome.go_rna_related.df$total <- rep(0, nrow(interactome.go_rna_related.df))
interactome.go_rna_related.df[i1,]$total <- wcl.df[i1,]$count
interactome.go_rna_related.df$source <- rep("GO_RNA_related", nrow(interactome.go_rna_related.df))
interactome.go_rna_related.df$ID <- rownames(interactome.go_rna_related.df)
interactome.go_rna_related.df$count <- rep(0, nrow(interactome.go_rna_related.df))
interactome.go_rna_related.df$frac <- rep(0, nrow(interactome.go_rna_related.df))

for (i in rownames(interactome.go_rna_related.df)) {
```

```
  kL1 <- keggLink("mmu", paste("mmu", i, sep = ""))
  interactome.go_rna_related.df[i, ]$count <- length(which(interactome.go_rna_related.keggIDs %in% kL1))
}

# extract list of IDs in pathway
interactome.go_rna_related.in_path.IDs <- lapply(rownames(interactome.go_rna_related.df), function(x) {
  kL1 <- keggLink("mmu", paste("mmu", x, sep = ""))
  in_path <- interactome.go_rna_related.keggIDs[which(interactome.go_rna_related.keggIDs %in% kL1)]
})
names(interactome.go_rna_related.in_path.IDs) <- rownames(interactome.go_rna_related.df)

# perform Fisher's Exact Test for each category
# Using WCL as background
bkgd <- length(unique(wcl.keggIDs))
smpl <- length(interactome.go_rna_related.keggIDs)

ftl <- apply(interactome.go_rna_related.df, 1, function (x) {
  ct <- as.integer(x["count"])
  tt <- as.integer(x["total"])
  m1 <- matrix(c(ct, tt, smpl - ct, bkgd - tt), 2, 2)
  fisher.test(m1, alternative = alternative)
})

interactome.go_rna_related.df$ft_pval <- unlist(lapply(ftl, function(x) {x$p.value}))
interactome.go_rna_related.df$ft_OR <- unlist(lapply(ftl, function(x) {x$estimate}))
interactome.go_rna_related.df$ft_fdr <- p.adjust(interactome.go_rna_related.df$ft_pval, method = p.adjus
save(interactome.go_rna_related.df, file = "data/interactome.go_rna_related.df")

# summarizing data at "B" level before doing Fisher's Exact test
interactome.go_rna_related.B.df <- data.frame(matrix(ncol = 5, nrow = length(unique(interactome.go_rna_
colnames(interactome.go_rna_related.B.df) <- c("B", "A", "total", "count", "source")
interactome.go_rna_related.B.df$B <- unique(interactome.go_rna_related.df$B)
interactome.go_rna_related.B.df$A <- sapply(unique(interactome.go_rna_related.df$B), function(x) {A <- u
interactome.go_rna_related.B.df$source <- rep("GO_RNA_related", nrow(interactome.go_rna_related.B.df))
interactome.go_rna_related.B.df$total <- sapply(unique(interactome.go_rna_related.df$B), function(x) {to
interactome.go_rna_related.B.df$count <- sapply(unique(interactome.go_rna_related.df$B), function(x) {co

ftl <- apply(interactome.go_rna_related.B.df, 1, function (x) {
  ct <- as.integer(x["count"])
  tt <- as.integer(x["total"])
  m1 <- matrix(c(ct, tt, smpl - ct, bkgd - tt), 2, 2)
  fisher.test(m1, alternative = alternative)
})

interactome.go_rna_related.B.df$ft_pval <- unlist(lapply(ftl, function(x) {x$p.value}))
interactome.go_rna_related.B.df$ft_OR <- unlist(lapply(ftl, function(x) {x$estimate}))
interactome.go_rna_related.B.df$ft_fdr <- p.adjust(interactome.go_rna_related.B.df$ft_pval, method = p.a
save(interactome.go_rna_related.B.df, file = "data/interactome.go_rna_related.B.df.rda")
```

Plotting the results of the enrichment analysis:

```
library(ggplot2)
library(grid)
```

```r
library(scales)
load("data/interactome.df.rda")
load("data/interactome.go_rna_unrelated.df.rda")
load("data/interactome.go_rna_related.df")


dfC <- rbind(interactome.df[, c("A", "B", "C", "ft_OR", "ft_fdr", "source", "count")],
             interactome.go_rna_related.df[, c("A", "B", "C", "ft_OR", "ft_fdr", "source", "count")],
             interactome.go_rna_unrelated.df[, c("A", "B", "C", "ft_OR", "ft_fdr", "source", "count")]
)

dfC$source <- as.factor(dfC$source)
dfC$source <- factor(dfC$source, levels = levels(dfC$source)[c(3,1,2)])

select1 <- unique(as.character(dfC[which(dfC$ft_fdr <= 0.1 & dfC$ft_OR > 1),]$C))
select1.pathIDs <- paste("mmu", unlist(lapply(strsplit(select1, "\\ "), function(x) x[1])), sep = "")
dfC <- dfC[which(dfC$C %in% select1),]

dfC$C <- as.factor(as.character(dfC$C))
dfC$ft_OR.cut <- cut(log2(dfC$ft_OR), breaks = c(-Inf,-4:4), right = F)
dfC$C <- factor(dfC$C, levels = levels(dfC$C)[dfC[dfC$source == "Interactome", "C"][order(dfC[which(dfC$

# formatting labels etc for plotting
l1 <- levels(dfC$ft_OR.cut)
l1 <- gsub("\\[", "", l1)
l1 <- gsub("\\)", "", l1)
levels(dfC$ft_OR.cut) <- l1

l1 <- as.character(levels(dfC$C))
l1 <- unlist(lapply(strsplit(l1, " "), function(x) {
  for (i in 2:length(x)){
    if (i == 2){
      v <- x[i]
    } else {
      v <- paste(v, x[i])
    }
  }
  return(v)
}))
levels(dfC$C) <- l1

levels(dfC$source)[2:3] <- c("RNA-related", "RNA-unrelated")


levels(dfC$C)[3] <- "Ribosome biogenesis"
levels(dfC$C)[6] <- "TCA cycle"
levels(dfC$C)[7] <- "mRNA surveillance"
levels(dfC$C)[11] <- "AA biosynthesis"
levels(dfC$C)[8] <- "H. simplex infection"
levels(dfC$C)[9] <- "Antibiotic biosynthesis"
levels(dfC$C)[12] <- "Glycolysis/Gluconeogenesis"

flevels <- levels(dfC$source)
```

```r
l1 <- factor(dfC$C, levels = levels(dfC$C)[c(1,2,3,7,4,5,6,12,8,9,10,11)])
dfC$C <- l1

p1 <- ggplot(data = dfC, aes(y = source, x = C)) +
      geom_tile(aes(fill = ft_OR.cut), colour = "white") +
      scale_fill_manual(values = brewer_pal(pal = "PuOr")(8), labels = levels(dfC$ft_OR.cut)) + #
      theme(axis.text.y = element_text(angle = 0, size = 8), axis.title = element_blank()) +
      guides(fill = guide_legend(label.position = "bottom", direction = "horizontal")) +
      theme(axis.text.x = element_text(angle = 45, vjust = 0.9, hjust = 0.8, size = 10)) +
      labs(fill = "Log2 OR") +
      scale_y_discrete(limits = rev(flevels)) +
      theme(legend.position = c(0.4,-1.92),
            legend.text = element_text(size = 4),
            legend.text.align = 0.5,
            legend.title = element_text(size = 4, vjust = 5),
            legend.key.size = unit(3.5, "mm"),
            legend.key.width = unit(3.5, "mm"),
            legend.margin = unit(0, "mm"),
            panel.margin = unit(1, "mm")) +
      ggtitle("KEGG pathway enrichment")

plot(p1)
```