RNA investigation - TLR expression across tissues

Find annotated TLR genes in ENSEMBL annotation

Genome used: gadmor3 ensembl Gadus_morhua.gadMor3.0.dna.toplevel.fa and Gadus_morhua.gadMor3.0.110.gtf

TLR reference sequences from https://pubmed.ncbi.nlm.nih.gov/27126702/ blasted towards gadmor3 using the ensembl web blast tool protein to DNA. Some genes have partial or missing gene models compared to manual curation e.g TLR21 and some TLR8.

Obtaining RNA data from different Atlantic cod tissues

In SRA, the following bio-project samples were selected for download:

sample	bio_project	strand	tissue	tissue_type
Ovary_SRR2045415	SRR2045415	unstranded	Ovary	fatty
Brain_SRR2045416	SRR2045416	unstranded	Brain	fatty
Gills_SRR2045417	SRR2045417	unstranded	Gills	bone
Heart_SRR2045418	SRR2045418	unstranded	Heart	muscle
Muscle_SRR2045419	SRR2045419	unstranded	Muscle	muscle
Liver_SRR2045420	SRR2045420	unstranded	Liver	fatty
Kidney_SRR2045421	SRR2045421	unstranded	Kidney	blood
Bones_SRR2045422	SRR2045422	unstranded	Bones	bone
Intestine_SRR2045423	SRR2045423	unstranded	Intestine	muscle
Testis_SRR2045424	SRR2045424	unstranded	Testis	fatty

The corresponding raw mRNA sequence data was downloaded and fastq extracted for analysis.

```
module purge
module load SRA-Toolkit/3.0.3-gompi-2022a

while read ACCESSION
do
prefetch ${ACCESSION}
fasterq-dump ${ACCESSION} -O
/gadiformes_genomes_rna/cod_rna_for_expression_test/rna_fastq_files -t /temp -e 12
done <list_sras
```

Mapping to reference and count extraction

```
module purge

module load STAR/2.7.10b-GCC-11.3.0

STAR \
--runThreadN 4 \
--runMode genomeGenerate \
--genomeDir . \
--genomeFastaFiles Gadus_morhua.gadMor3.0.dna.toplevel.fa \
--sjdbGTFfile Gadus_morhua.gadMor3.0.110.gtf \
1>gadmor3_index.out 2>gadmor3_index.err

##


while read R1 R2 R3
do

mkdir ${R3}_ensembl
cd ${R3}_ensembl

STAR \
--runThreadN 6 \
--runMode alignReads \
--genomeDir /gadmor3_genome_star_ensembl \
--sjdbGTFfile /gadmor3_genome_star_ensembl/Gadus_morhua.gadMor3.0.110.gtf \
--readFilesIn \
/cod_rna_for_expression_test/rna_fastq_files/${R1} \
/cod_rna_for_expression_test/rna_fastq_files/${R2} \
--outSAMtype BAM SortedByCoordinate \
--quantMode GeneCounts \
--outFileNamePrefix ${R3}_ \
1>${R3}_mapping.out 2>${R3}_mapping.err

cd cod_rna_for_expression_test

done </cod_rna_for_expression_test/rna_fastq_files/list_rna_input
```

Count analysis

As there are no replicates, and these samples are unrelated, **the overall goal of the below code is to demonstrate present TLR expression across different tissues without any absolute or relative quantification**. The unstranded raw counts were merged into a count matrix and read into RStudio. In R, utilising some functions in edgeR and DESeq, the count data was converted to log2 transformed counts per million (cpm) for boxplots and log2 transformed counts for single-gene expression plots. These plots are meant as proof of TLR expression across tissues only. Comparison between tissues is not possible for the current setup.

```
# remove headers in count files

for i in $(ls *ReadsPerGene.out.tab)
do
sed -e '1,4d' ${i} > ${i}_clean
done

# extract counts for unstranded
for i in $(ls *Gene.out.tab_clean)
do
awk '{print $1,$2}' ${i} > ${i}_final_unstranded
done

# add new header

sed -i '1iID Ovary_SRR2045415'
SRR2045415_ReadsPerGene.out.tab_clean_final_unstranded
sed -i '1iID Brain_SRR2045416'
SRR2045416_ReadsPerGene.out.tab_clean_final_unstranded
sed -i '1iID Gills_SRR2045417'
SRR2045417_ReadsPerGene.out.tab_clean_final_unstranded
sed -i '1iID Heart_SRR2045418'
SRR2045418_ReadsPerGene.out.tab_clean_final_unstranded
sed -i '1iID Muscle_SRR2045419'
SRR2045419_ReadsPerGene.out.tab_clean_final_unstranded
sed -i '1iID Liver_SRR2045420'
SRR2045420_ReadsPerGene.out.tab_clean_final_unstranded
sed -i '1iID Kidney_SRR2045421'
SRR2045421_ReadsPerGene.out.tab_clean_final_unstranded
sed -i '1iID Bones_SRR2045422'
SRR2045422_ReadsPerGene.out.tab_clean_final_unstranded
sed -i '1iID Intestine_SRR2045423'
SRR2045423_ReadsPerGene.out.tab_clean_final_unstranded
sed -i '1iID Testis_SRR2045424'
SRR2045424_ReadsPerGene.out.tab_clean_final_unstranded

# make matrix

paste *ReadsPerGene.out.tab_clean_final_unstranded | awk '{print $1 , $2 , $4 , $6
, $8 , $10, $12, $14, $16, $18, $20}' > codRNA_ensembl.unstranded.matrix
```

```
library("edgeR")
library("ggplot2")
library("dplyr")
library("DEFormats")
library("DESeq2")

sessionInfo()
# R version 4.3.2 (2023-10-31 ucrt)
# Platform: x86_64-w64-mingw32/x64 (64-bit)
# Running under: Windows 10 x64 (build 19045)
#
# Matrix products: default
#
# locale:
# [1] C
#
# time zone: Europe/Paris
# tzcode source: internal
#
# attached base packages:
# [1] stats4    stats     graphics  grDevices utils     datasets  methods   base
#
# other attached packages:
#  [1] reshape2_1.4.4              DESeq2_1.40.2
SummarizedExperiment_1.30.2 Biobase_2.60.0
#  [5] MatrixGenerics_1.12.3       matrixStats_1.0.0
GenomicRanges_1.52.0        GenomeInfoDb_1.36.3
#  [9] IRanges_2.34.1              S4Vectors_0.38.1            BiocGenerics_0.46.0
DEFormats_1.28.0
# [13] dplyr_1.1.3                 ggplot2_3.4.3               edgeR_3.42.4
limma_3.56.2
#
# loaded via a namespace (and not attached):
#  [1] utf8_1.2.3              generics_0.1.3          bitops_1.0-7
stringi_1.7.12          lattice_0.21-9
#  [6] magrittr_2.0.3          grid_4.3.2              plyr_1.8.8
Matrix_1.6-1            backports_1.4.1
# [11] fansi_1.0.4             scales_1.3.0            codetools_0.2-19
abind_1.4-5             cli_3.6.1
# [16] rlang_1.1.1             crayon_1.5.2            XVector_0.40.0
munsell_0.5.0           DelayedArray_0.26.7
# [21] withr_2.5.2             S4Arrays_1.0.6          tools_4.3.2
parallel_4.3.2          BiocParallel_1.34.2
# [26] checkmate_2.3.1         colorspace_2.1-0        locfit_1.5-9.8
GenomeInfoDbData_1.2.10 vctrs_0.6.3
# [31] R6_2.5.1                lifecycle_1.0.4         stringr_1.5.1
zlibbioc_1.46.0         pkgconfig_2.0.3
# [36] pillar_1.9.0            gtable_0.3.4            data.table_1.14.8
glue_1.6.2              Rcpp_1.0.11
# [41] tibble_3.2.1            tidyselect_1.2.0        rstudioapi_0.15.0
farver_2.1.1            labeling_0.4.3
# [46] compiler_4.3.2          RCurl_1.98-1.12
```

Data read-in

```
# reading in unstranded count data, metainformation and annotation
```

```
annotations <- read.table("TLR_IDs.txt", sep="\t", header=T)

#ENSEMBL_ID GENE_NAME
#ENSGMOG00000003793 TLR14
#ENSGMOG00000000110 TLR22
#ENSGMOG00000023329 TLR22
#ENSGMOG00000023416 TLR22
#ENSGMOG00000024474 TLR22
#ENSGMOG00000027403 TLR22
#ENSGMOG00000030395 TLR22
#ENSGMOG00000032059 TLR22
#ENSGMOG00000033196 TLR22
#ENSGMOG00000023851 TLR23
#ENSGMOG00000028267 TLR23
#ENSGMOG00000024180 TLR25
#ENSGMOG00000024954 TLR25
#ENSGMOG00000027725 TLR25
#ENSGMOG00000032371 TLR25
#ENSGMOG00000033145 TLR25
#ENSGMOG00000033277 TLR25
#ENSGMOG00000035479 TLR25
#ENSGMOG00000036237 TLR25
#ENSGMOG00000035330 TLR3
#ENSGMOG00000031937 TLR7
#ENSGMOG00000037181 TLR7
#ENSGMOG00000024345 TLR8
#ENSGMOG00000028618 TLR8
#ENSGMOG00000030100 TLR8
#ENSGMOG00000033206 TLR8
#ENSGMOG00000036041 TLR8
#ENSGMOG00000003161 TLR9
#ENSGMOG00000003222 TLR9
#ENSGMOG00000011256 TLR9
#ENSGMOG00000024698 TLR9

counts.unstranded <- read.table("codRNA_ensembl.unstranded.matrix", header= TRUE,
sep =" ", row.names=1)

colData.all <- read.table("colData_cod_rna.txt", header=TRUE, sep="\t",
row.names=1)

dim(annotations)
dim(counts.unstranded)
dim(colData.all)

# 10 samples, 29063 gene regions, 31 annotated TLR genes in Ensembl
```

Data order verification

```
row.names(colData.all) == colnames(counts.unstranded)
```

edgeR - setting up the DGElist object

```
# setting preliminary group of interest - tissue. this is an arbitrary grouping.
All samples are unrelated and without replicates.

group=as.factor(colData.all$tissue)

#levels(group)
#Bones Brain Gills Heart Intestine Kidney Liver Muscle Ovary Testis

# make DGEList object

y_data.unstranded <- DGEList(counts=counts.unstranded, group=group)

#converting to CPM and log2 CPM
#y_data.unstranded.raw.cpm <- cpm(y_data.unstranded)
y_data.unstranded.raw.lcpm <- cpm(y_data.unstranded, log=TRUE)
```

Filtering on expression using tissue as group

```
keep_edgeR.unstranded <- filterByExpr(y_data.unstranded)

y_data.unstranded <-y_data.unstranded[keep_edgeR.unstranded, ,
keep.lib.sizes=FALSE]

dim(y_data.unstranded$counts)

#[1] 22326    10

#converting to CPM and log2 CPM
#y_data.unstranded.filt.cpm <- cpm(y_data.unstranded)
y_data.unstranded.filt.lcpm <- cpm(y_data.unstranded, log=TRUE)
```

Adding gene information

```
# note that the order and number of annotation/gene id in y_data is not the same
as in annotations. Therefore using function match:
```

```
m <- match(row.names(y_data.unstranded$counts), annotations$ENSEMBL_ID)

y_data.unstranded$genes <- data.frame(annotations$GENE_NAME[m])

# subsetting matrix for TLR identifiers only

counts_TLR_raw <- y_data.unstranded.raw.lcpm[row.names(y_data.unstranded.raw.lcpm)
%in% annotations$ENSEMBL_ID,]

counts_TLR_filt <-
y_data.unstranded.filt.lcpm[row.names(y_data.unstranded.filt.lcpm) %in%
annotations$ENSEMBL_ID,]

dim(counts_TLR_raw)
dim(counts_TLR_filt)

# 5 TLR genes were removed due to low expression across all samples
```

edgeR - count distributions

```
# all counts filtered

#par(mar=c(12.1, 4.1, 4.1, 2.1))
#boxplot(y_data.unstranded.filt.lcpm, xlab="", ylab="Log counts per
million",las=2,outline=FALSE)
#abline(h=median(y_data.unstranded.filt.lcpm),col="blue")
#title("Boxplots of logCPMs")

# unfiltered TLR LOCs log2 CPM

pdf("raw_TLR_expression_boxplot.pdf")
par(mar=c(12.1, 4.1, 4.1, 2.1))
boxplot(counts_TLR_raw, xlab="", ylab="Log2 counts per
million",las=2,outline=FALSE)
abline(h=median(counts_TLR_raw),col="blue")
abline(h=0,col="black", lty = "dashed")
title("Boxplots of unfiltered TLR log2CPMs")
dev.off()

# filtered TLR LOCs log2 CPM

pdf("filtered_TLR_expression_boxplot.pdf")
par(mar=c(12.1, 4.1, 4.1, 2.1))
boxplot(counts_TLR_filt, xlab="", ylab="Log2 counts per
million",las=2,outline=FALSE)
abline(h=median(counts_TLR_raw),col="blue")
abline(h=0,col="black", lty = "dashed")
title("Boxplots of TLR log2CPMs filtered by expression")
```

```
dev.off()
```

edgeR - normalization

```
# Library normalization

y_data.unstranded  <- normLibSizes(y_data.unstranded)
```

Single gene raw count plotting

Converting the edgeR object to a DESeq2 object to enable use of the raw count plot function called plotCounts.

```
dds_y_data.unstranded = as.DESeqDataSet(y_data.unstranded)

# loop through all TLR - note that a few TLR genes were lost in filtering. New
annotation list made:

counts_TLR_filt <-
y_data.unstranded.filt.lcpm[row.names(y_data.unstranded.filt.lcpm) %in%
annotations$ENSEMBL_ID,]

nred <- row.names(counts_TLR_filt)

annotations_reduced <- annotations[annotations$ENSEMBL_ID %in% c(nred),]

head(annotations_reduced)
#           ENSEMBL_ID GENE_NAME
#1 ENSGMOG00000003793     TLR14
#2 ENSGMOG00000000110     TLR22
#3 ENSGMOG00000023329     TLR22
#4 ENSGMOG00000023416     TLR22
#5 ENSGMOG00000024474     TLR22
#6 ENSGMOG00000027403     TLR22

for (row in 1:nrow(annotations_reduced)) {
    GENE <- annotations_reduced[row, "ENSEMBL_ID"]
    NAME  <- annotations_reduced[row, "GENE_NAME"]
    png(paste0("Log2_normalized_counts_",GENE,"_",NAME,".png"), width = 11, height
= 8, units = "in", res = 300)
    d <- plotCounts(dds_y_data.unstranded, gene=GENE, intgroup=c("group"),
returnData=TRUE)
    print(ggplot(d, aes(x=group, y=count)) +
geom_point(aes(color=colData.all$tissue), size=3) + ggtitle(paste0(NAME," ",GENE," - Log2 normalized counts")))
    dev.off()
}




for (row in 1:nrow(annotations)) {
    GENE <- annotations[row, "ENSEMBL_ID"]
    NAME  <- annotations[row, "GENE_NAME"]
    png(paste0("Log2_normalized_counts_",GENE,"_",NAME,"unfiltered.png"), width = 11, height = 8, units = "in", res = 300)
    d <- plotCounts(dds_y_data.unstranded, gene=GENE, intgroup=c("group"), returnData=TRUE)
    print(ggplot(d, aes(x=group, y=count)) + geom_point(aes(color=colData.all$tissue), size=3) + ggtitle(paste0(NAME," ",GENE," - unfiltered Log2 normalized counts")))
    dev.off()
}
```