RNA investigation - is MHCI expressed globally in most tissues

Find annotated MHCI genes in RefSeq annotation

Genome used: gadmor3 refseq GCF_902167405.1 with corresponding annotation.

MHC references from Grimholt et al 2015 (DOI: 10.1186/s12862-015-0309-1) were used as queries towards predicted nucleotide CDS from GCF_902167405.1 using TblastN.

```
module purge
module load BLAST+/2.14.1-gompi-2023a

tblastn \
-db Gadus_morhua.gadMor3.0.cds.all.fa \
-query Atlantic_cod_MHCI_for_blast_prot.fas \
-out blastout_ensembl_cds_mhci_tab \
-outfmt 6 \
-num_threads 10 \
-evalue 1e-5
```

All transcript hits were extracted, translated to protein and aligned together with the reference sequences using MEGA7. A few transcripts were removed due to gene model fusion. The alignment was subjected to a quick NJ analysis using Poisson distribution and pairwise deletion. Only transcripts grouping with MHCI U and Z lineage were noted. All corresponding LOC gene identifiers were extracted from GCF_902167405.1 RefSeq GTF file and noted as either MHCI U or Z.

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

sample	bio_project	strand	tissue	tissue_type
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 GCF_902167405.1_gadMor3.0_genomic.fna \
--sjdbGTFfile genomic.gtf \
1>gadmor3_index.out 2>gadmor3_index.err

module purge
module load STAR/2.7.10b-GCC-11.3.0

while read R1 R2 R3
do

#mkdir ${R3}
cd ${R3}

STAR \
--runThreadN 6 \
--runMode alignReads \
--genomeDir /gadiformes_genomes_rna/gadmor3_genome_star \
--sjdbGTFfile /gadiformes_genomes_rna/gadmor3_genome_star/genomic.gtf \
--readFilesIn \
/gadiformes_genomes_rna/cod_rna_for_expression_test/rna_fastq_files/${R1} \
/gadiformes_genomes_rna/cod_rna_for_expression_test/rna_fastq_files/${R2} \
--outSAMtype BAM SortedByCoordinate \
```

```
--quantMode GeneCounts \
--outFileNamePrefix ${R3}_ \
1>${R3}_mapping.out 2>${R3}_mapping.err

cd /gadiformes_genomes_rna/cod_rna_for_expression_test

done
</gadiformes_genomes_rna/cod_rna_for_expression_test/rna_fastq_files/list_rna_inpu
t
```

Count analysis

As there are no replicates, and these samples are unrelated, **the overall goal of the below code is to demonstrate present MHCI 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 MHCI expression across tissues only. Comparison between tissues is not possible for the current setup.

```
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("MHCI_LOC_IDs.txt", sep="\t", header=T)

counts.unstranded <- read.table("codRNA.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, 32848 gene regions, 54 annotated MHCI U and Z lineage genes in
RefSeq
```

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] 23353    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$counts), annotations$LOC)

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

# subsetting matrix for MHCI identifiers only

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

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

dim(counts_MHCI_raw)
dim(counts_MHCI_filt)

# 3 MHCI genes were removed due to low expression across all samples

#counts_MHCI_raw.cpm <-
y_data.unstranded.raw.cpm[row.names(y_data.unstranded.raw.cpm) %in%
annotations$LOC,]

#counts_MHCI_filt.cpm <-
y_data.unstranded.filt.cpm[row.names(y_data.unstranded.filt.cpm) %in%
annotations$LOC,]
```

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 MHCI LOCs log2 CPM

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

```
# filtered MHCI LOCs log2 CPM

pdf("filtered_MHCI_expression_boxplot.pdf")
par(mar=c(12.1, 4.1, 4.1, 2.1))
boxplot(counts_MHCI_filt, xlab="", ylab="Log2 counts per
million",las=2,outline=FALSE)
abline(h=median(counts_MHCI_raw),col="blue")
abline(h=0,col="black", lty = "dashed")
title("Boxplots of MHCI log2CPMs filtered by expression")
dev.off()

# unfiltered MHCI LOCs CPM

#par(mar=c(12.1, 4.1, 4.1, 2.1))
#boxplot(counts_MHCI_raw.cpm, xlab="", ylab="counts per
million",las=2,outline=FALSE)
#abline(h=median(counts_MHCI_raw.cpm),col="blue")
#title("Boxplots of raw MHCI CPMs")

# filtered MHCI LOCs CPM

#par(mar=c(12.1, 4.1, 4.1, 2.1))
#boxplot(counts_MHCI_filt.cpm, xlab="", ylab="counts per
million",las=2,outline=FALSE)
#abline(h=median(counts_MHCI_filt.cpm),col="blue")
#title("Boxplots of edgeR filtered by expression MHCI CPMs")
```

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 MHCI - note that a few MHCI genes were lost in filtering. New
annotation list made:
```

```
counts_MHCI_filt <-
y_data.unstranded.filt.lcpm[row.names(y_data.unstranded.filt.lcpm) %in%
annotations$LOC,]

nred <- row.names(counts_MHCI_filt)

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

head(annotations_reduced)
#           LOC  GENE
#1 LOC115529270 MHCIU
#2 LOC115529271 MHCIU
#3 LOC115529272 MHCIU
#4 LOC115529274 MHCIU
#5 LOC115529275 MHCIU
#6 LOC115529323 MHCIU

for (row in 1:nrow(annotations_reduced)) {
    GENE <- annotations_reduced[row, "LOC"]
    NAME  <- annotations_reduced[row, "GENE"]
    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()
}
```