title: "Project_main" output: pdf_document: default

html_document: default

Introduction

Recent decades have seen a rapid increase in global warming and the Arctic is no exception. In response to the warming of the Arctic, marine boreal species are expanding into the Arctic at a pace that reflects environmental warming. Hence, several marine species have overspread the Arctic, and the establishment of these species has resulted in changes in regimes, assemblage compositions, and food web structures. However, the range expansion of boreal marine species is constrained by the ability of these species to tolerate low water temperatures and subzero air temperatures during winter from intertidal species. The temperature rises above the normal range are beyond the thermoregulatory capacity of many boreal intertidal species, such as the highly abundant blue mussel Mytilus edulis, which cannot survive at 32 °C despite repeated exposures to air.

Physicochemical and ecological conditions modify the body temperature of intertidal organisms during low tide, making it difficult to predict heat stress in these organisms. Therefore, remotely sensed atmospheric air temperatures cannot capture physiological conditions at the microhabitat level. For a deeper understanding of the role of thermal stress on natural populations, field-based physiological and molecular evaluations are required. Laboratory studies on thermal stress are prone to overestimating the actual impact of global warming on the physiology of intertidal organisms. In the current study, in order to increase knowledge regarding intertidal heat stress in the Arctic, a series of experiments has been conducted to identify if current Arctic summer temperatures induce any in situ cellular stress responses in Greenland Mytilus edulis. At the first step, across a natural temperature gradient, samples from the first warm days of the year were compared with those taken at similarly manipulated temperatures. Accordingly, Mytilus edulis were collected from the inner (warmer) and outer (cooler) regions of the Godthåbsfjorden around Nuuk (64°N) in order to study the effect of the fjord temperature gradient. Then, collected samples were exposed to two acute temperature shocks: 22°C and 32°C, which represent common and extreme summer air temperatures for intertidal habitats in Nuuk.

Materials and methods

All software and packages used were managed under environment control with conda.

The files were downloaded from https://www.ebi.ac.uk/ and where insufficient links were provided, a script (scripts/scp_ftps.sh in the Github repo) was created to obtain them, see Supplementary information for more instructions on this. Quality control was done for all 48 (24 paired end reads) samples with fastqc. A multiQC was made to obtain a single HTML file for all samples.

```
# install fastqc using conda
conda install -c bioconda fastqc
# performing fastqc for all fastq files
fastqc -o /home/student5/results/fastqc_report /home/student6/data/*.fastq
# install multiqc using conda
conda install -c bioconda -c conda-forge multiqc
# performing multiqc
multiqc .
open multiqc_report.html
# or open multiqc.html file in browser
scp student5@13.48.179.168:/home/student5/results/fastqc_report/multiqc_report.html ./Desktop
# performing fastqc for MC2_1.fastq (for comparing the result from one of the fastq files with Supplementary Table S1)
fastqc /home/student5/Documents/MC2_1.fastq
# open html file in browser
scp student5@13.48.179.168:/home/student5/Documents/MC2_1_fastqc.html ./Desktop
```

De novo transcriptome assembly was conducted using Trinity version 2.13.1. For Trinity to work properly the data files needed to be unziped which was done by running the script in scripts/unzip-files.sh as follows:

```
./unzip-files.sh <PATH-TO-FOLDER-WITH-FILES>
```

Trinity was run for two transcriptomes on their provided provided Docker image using Singularity as follows:

```
singularity build trinity.simg docker://trinityrnaseq/trinityrnaseq:2.13.1
nohup singularity exec -e trinity.simg Trinity --seqType fq \
      --left `pwd`/data/ERR4859139_1.fastq --right `pwd`/data/ERR4859139_2.fastq \
      --max_memory 6G --CPU 4 --output `pwd`/trinity_out > nohup.out &
```

However, due to extensive time consumption running the Trinity assembly, all samples could not be processed.

The assembled transcriptome for the control animal was used as reference five 4 samples, one from each condition was choosen to be mapped into the transcriptome. Minimap2 was to align the samples to the assembled transcriptome. The sam-files were converted to bam-files using samtools to be sorted and loaded faster. An indexing was also done to be able to look at the alignments in Integrative Genome Viewer.

```
#conda activate minimap2
minimap2 -ax sr bwa/MC1.Trinity.fasta MI5_1.fastq MI5_2.fastq > MI5_alignment.sam
samtools view -b MO1_alignment.sam > MO1_alignment.bam
samtools sort MO1_alignment.bam > MO1_alignment_sorted.bam
samtools index MO1_alignment_sorted.bam
```

The sam-file was now to be converted into counts, but due to lack of a GFF-file and lack of time to create one from the assembly this was not feasible.

Since the transcriptomes could not be used the way desired, the same four samples was again choosen to do the possible downstream analysis on. The paired end read files were merged to single files using micca - MICrobial Community Analysis, again from a Docker image as follows:

```
singularity build micca.simg docker://compmetagen/micca:1.7.2
nohup singularity exec micca.simg micca mergepairs -i \
          [`pwd`/data/MC5_1.fastq `pwd`/data/MC5_2.fastq] \
          -o `pwd`/data-merged/MC5_paired.fastq > nohup-micca.out &
```

Where MC5_1.fastq and MC5_2.fastq are the pair end reads for one sample. The merged files were then converted from fastq to fasta files using the fastx toolbox.

```
conda activate filter #here I have the fastx_toolbox
cd home/student6/data-merged/
#ex:
fastq_to_fasta -i MC5_paired.fastq -o MC5_paired.fasta
```

Orfipy was then used to find the coding ORFs of the samples setting gene lengths between 201 and 30590 in an attempt to reproduce the findings of unigenes from original paper, using another method.

```
# conda create -n orfipy
# conda activate orfipy
# conda install -c bioconda orfipy
orfipy MO1_paired.fasta --dna ORFS_MO1.fa --min 201 --max 30590
# since we had to do this step in another way, we set the min and max values of
#the ORFs to the values that they found
#then I put all the ORF_id.fa files in ORFS directory
```

The found ORFs were transcribed from DNA into amino acid sequence using transeq in the emboss package.

```
conda create -n emboss
conda install -c bioconda emboss
conda activate emboss
transeq ORFS_MC5.fa ORFS_MC5.aa #making the aa files
```

A BLASTP search was done against SwissProt darabase.

```
#moving all to a new directory
mkdir gene_predicition
mv ORFS/*.aa gene_prediction/ #moving all files with ending .aa
#in the gene_prediction directory
conda activate gene_prediction
wget https://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.fasta.gz
gunzip uniprot_sprot.fasta.gz
makeblastdb -in uniprot_sprot.fasta -dbtype prot -out uniprot_database
#then i run the blast p for each file..
blastp -query ORFS_MC5.aa -db uniprot_database -outfmt 7 -out blastp_MC5
```

The genes found in all samples were then counted.

```
grep -oP '(?<=\|)[^\|]+' blastp_S323 |awk 'NR % 2 == 1' > S323_all_prot
# awk 'NR % 2 == 1' prints all uneven lines
#do for all files
cat MC5_all_prot MI5_all_prot MO1_all_prot S323_all_prot > all_prot
sort -u all_prot | wc -l # --> 290122 genes
```

The files were then filtered using a cutoff for the e-value of 0.00001, only the genes annotated to Mytilus edulis species were choosen and the gene count for each sample was counted.

```
# e-value cutoff set to e^-5 and code for species = MYTED
grep "^[^#;]" blastp_MC5 | awk '$11 <= 0.00001 {print$0}' | awk '/MYTED/ {print}' > filtered_blastp_MC5
grep "^[^#;]" blastp_MO1 | awk '$11 <= 0.00001 {print$0}' | awk '/MYTED/ {print}' > filtered_blastp_MO1
grep "^[^#;]" blastp_MI5 | awk '$11 <= 0.00001 {print$0}' | awk '/MYTED/ {print}' > filtered_blastp_MI5
grep "^[^#;]" blastp_S323 | awk '$11 <= 0.00001 {print$0}' | awk '/MYTED/ {print}' > filtered_blastp_S323
grep "^[^#;]" blastp_S221 | awk '$11 <= 0.00001 {print$0}' | awk '/MYTED/ {print}' > filtered_blastp_S221
```

```
#protein names extraction
grep -oP '(?<=\|)[^\|]+' filtered_blastp_MC5 |awk 'NR % 2 == 1' > onlyprot_MC5
wc -l onlyprot_MC5 # --> 5771
grep -oP '(?<=\|)[^\|]+' filtered_blastp_MI5 |awk 'NR % 2 == 1' > onlyprot_MI5
wc -l onlyprot_MI5 # --> 13209
grep -oP '(?<=\|)[^\|]+' filtered_blastp_MO1 |awk 'NR % 2 == 1' > onlyprot_MO1
wc -l onlyprot_MO1 # --> 1562
grep -oP '(?<=\|)[^\|]+' filtered_blastp_S323 |awk 'NR % 2 == 1' > onlyprot_S323
wc -l onlyprot_S323 # --> 3386
```

Results and discussion

Quality control of each sample showed that the quality of the samples were very high with phred scores of over 25 for all samples. Thus, an assumption that the samples were already filtered and the analysis was proceeded without any further edits of the files.

A crucial complication with the reproduction of this study was the large amount of time and computational power required for the assembly of the transcriptomes. Since this step could not be done to satisfaction, all downstream analysis was obstructed.

The transcriptome obtained from Trinity that was used as a reference, did not have a respective GFF/annotation file. With further research on how to obtain one from Trinity as well as with more time, this step could possibly have been solved. However, within the given time frame this was not feasible for our level of knowledge. Also, it was realised too late in the procedure that we needed a GFF/annotation file to convert the sam-file into counts and thus, the desired DE gene analysis and GO-term enrichment analysis could not be done. As there were also no exisitng files in Ensembl or RefSeq (NCBI), to redo the aligmnent with a genome was also not an option.

Further, since all bioanalytical analyses was made by a company, the details about the analysis done was documented only to a small extent. Therefore, a large amount of packages were tested before reaching the code that gave the results.

In the original study 402 060 unigenes were found and our, somewhat improvised, corresponding number was calculated to 290 122. This number was obtained from the blast search with all genes in the whole SwissProt database containing a large amount of different species. This number is therefore not realistic, since the blue mussel most likely not expresses genes from other species. Further, the 402 060 unigenes the original paper claim to have found is also not a realistic number. Given that the human genome consists of approximately 25 000 to 30 000 genes, we find it highly unlikely that the blue mussel has over 400 000 genes.

The blast search was also filtered to obtain only the genes with sufficient e-value and also only the genes coupled to Mytilus edulis species. The number of genes obtained for each sample can be found in Supplementary information. We observe a great variance between the samples, especially for MI5, and this is likely because this sample had more reads and therefore more possible hits. No conclusions can be drawn from these single samples from each condition, however, the number of genes of 2000-13000 genes, seems like a more reasonable number of genes compared to the 400 000 unigenes found in the original article.