The following supplement accompanies the article

Feeding by Antarctic krill *Euphausia superba* in the West Antarctic Peninsula: differences between fjords and open waters

Alison C. Cleary*, Edward G. Durbin, Maria C. Casas

*Corresponding author: alison.cleary@npolar.no

Marine Ecology Progress Series 595: 39–54 (2018)

```
#!/bin/bash
#SBATCH -t 10:00:00
#SBATCH --mem=50G
#SBATCH -n 16
#SBATCH -J Krill_analysis

#This script is used to process paired-end illumina sequence data for the analysis of krill gut contents and is provided as supplementary material with Cleary et al. (2018)
module load qiime/1.9.0
module load oligotyping
module load blast

#unzip files
cd data/acleary/Krill/All_samples/;
gunzip *.fastq.gz;
cd;

cd data/acleary/

#make a directory to keep the analyses together
mkdir Krill/2017/

#Join paired ends of all files, note that the parameters file specifies a minimum overlap of 50 base pairs, and a maximum of 10% difference across the overlap region
multiple_join_paired_ends.py -i Krill/All_samples/ -o Krill/2017/all_krill_paired/ -p scripts/parameters.txt

#move the unjoined files elsewhere so they do not get joined in with split libraries
mkdir Krill/2017/unpaired_reads/
find Krill/2017/all_krill_paired/ -name "fastqjoin.un*" -print -exec mv {} Krill/2017/unpaired_reads/ \;

#add Qiime labels to designate sample, and combine all reads into one file
multiple_split_libraries_fastq.py -i Krill/2017/all_krill_paired/ -o Krill/2017/all_krill_split_libraries/ --include_input_dir_path

#remove reverse primers, then flip the sequences and remove the forward primers. The -z flag removes any reads which do not contain both of the correct primer sequences, as these sequences are likely to be of low quality. Leave the seqs in this reversed orientation b/c this is now true 5'->3' (due to the reverse primer being associated with the R1 adaptor and vice versa)
truncate_reverse_primer.py -f Krill/2017/all_krill_split_libraries/seqs.fna -m Krill/maps/Map_F_primering3.txt -o Krill/2017/all_krill_split_libraries/first_trimmed/ -z truncate_remove;
```

```
adjust_seq_orientation.py -i
Krill/2017/all_krill_split_libraries/first_trimmed/seqs_rev_primer_truncated.fna -o
Krill/2017/all_krill_split_libraries/first_trimmed/seqs_rev_primer_truncated_RC.fna -r

truncate_reverse_primer.py -f
Krill/2017/all_krill_split_libraries/first_trimmed/seqs_rev_primer_truncated_RC.fna -m
Krill/maps/Map_R_primering3.txt -o Krill/2017/all_krill_clean_reads/ -z truncate_remove;

#This is a large data set, which causes it to run into memory limits in the Oligotyping. In order to optimize
oligotyping, we remove as many "bad" sequences as possible with a rough first cut before the oligotyping.
A rough cut 95% OTUs is used for this, and then sequences which are krill, or singletons, are removed.

#Make 95% OTUs with UCLUST

mkdir Krill/2017/filtering/

pick_otus.py -i
Krill/2017/all_krill_clean_reads/seqs_rev_primer_truncated_RC_rev_primer_truncated.fna  -o
Krill/2017/filtering/95p_otus/ -s 0.95

#pick representative sequences for each of the rough cut OTUs
pick_rep_set.py -i
Krill/2017/filtering/95p_otus/seqs_rev_primer_truncated_RC_rev_primer_truncated_otus.txt -f
Krill/2017/all_krill_clean_reads/seqs_rev_primer_truncated_RC_rev_primer_truncated.fna -m
most_abundant -o  Krill/2017/filtering/95p_otus/95p_otu_rep_set.fna

#assign taxonomy to each of the rough cut OTUs by automated comparison against the Silva database
assign_taxonomy.py -i Krill/2017/filtering/95p_otus/95p_otu_rep_set.fna -o
Krill/2017/filtering/95p_otus/taxonomy2/ -t reference_sets/99_Silva_111_taxa_map.txt -r
reference_sets/99_Silva_111_rep_set.fasta

#Make a list of singletons by selecting any OTU which only has information in 2 fields (denovoXX and
one sequence), and trim to just the first column to make a list of singleton OTU names
awk -F'\t' 'NF==2'
Krill/2017/filtering/95p_otus/seqs_rev_primer_truncated_RC_rev_primer_truncated_otus.txt  >
Krill/2017/filtering/95p_otus/singletons_otus.txt

cut –f1 Krill/2017/filtering/95p_otus/singletons_otus > Krill/2017/filtering/95p_otus/singletons_list.txt

#Preliminary results showed no otus of non-krill malacostraca above 1000 reads. Therefore, to simplify
and automate removing krill reads, all malacostraca reads are removed. This works better than removing
only otus identified as euphausia because it captures many of the noisy, poor quality reads, which are
only assigned to this higher taxonomic level

#first grab all the lines of the taxa map file which do not contain "Malacostraca"
grep -F -v Malacostraca
Krill/2017/filtering/95p_otus/taxonomy2/95p_otu_rep_set_tax_assignments.txt >
Krill/2017/filtering/95p_otus/non_krill_otus_tax_assignments.txt

#Now cut out just the otu number
cut -f1 Krill/2017/filtering/95p_otus/non_krill_otus_tax_assignments.txt >
Krill/2017/filtering/95p_otus/non_krill_otus.txt

#Now remove singletons, the -w option is important to match only whole words (otherwise e.g. denovo1
```

```
also removes denovo100, denovo1123, etc.)
grep -F -v -w -f Krill/2017/filtering/95p_otus/singletons_list.txt
Krill/2017/filtering/95p_otus/non_krill_otus.txt > Krill/2017/filtering/95p_otus/otus_to_keep.txt

#Next use this list from the tax assignments to grab the list of corresponding reads from the otu map
grep -w -F -f Krill/2017/filtering/95p_otus/otus_to_keep.txt
Krill/2017/filtering/95p_otus/seqs_rev_primer_truncated_RC_rev_primer_truncated_otus.txt >
Krill/2017/filtering/95p_otus/seqs_to_keep_otus.txt

#Next cut out the first column (which contains the OTU names) so you are left with just the sequence
identifiers
cut -f2- Krill/2017/filtering/95p_otus/seqs_to_keep_otus.txt >
Krill/2017/filtering/95p_otus/seqs_to_keep.txt

#Now change all the tabs to carriage returns to make it into a real list
 tr '\t' '\n' < Krill/2017/filtering/95p_otus/seqs_to_keep.txt >
Krill/2017/filtering/95p_otus/seqs_to_keep_list.txt

#Now use this list of sequences to make a new full fasta file of "good" paired-end reads
grep -F -w -A 1 -f Krill/2017/filtering/95p_otus/seqs_to_keep_list.txt
Krill/2017/all_krill_clean_reads/seqs_rev_primer_truncated_RC_rev_primer_truncated.fna >
Krill/2017/Seq_reads_no_krill_nor_singletons.fna

#make a directory to put oligotyping results in
mkdir Krill/2017/oligotyping/

#pick oligotypes
o-pad-with-gaps Krill/2017/Seq_reads_no_krill_nor_singletons.fna;
mv Krill/2017/Seq_reads_no_krill_nor_singletons.fna-PADDED-WITH-GAPS
Krill/2017/oligotyping/seqs_padded_with_gaps.fna;

sed -i 's/ .*$//g'  Krill/2017/oligotyping/seqs_padded_with_gaps.fna;

decompose Krill/2017/oligotyping/seqs_padded_with_gaps.fna -o Krill/2017/oligotyping/ -S -M
100 --skip-gen-figures --skip-check-input-file --skip-gexf-files;

#get rid of all the tailing gaps from the oligotype rep set so that it is ready for taxonomic
assignment
sed 's/-//g' Krill/2017/oligotyping/NODE-REPRESENTATIVES.fasta >
Krill/2017/oligotyping/18S_node_representatives.fna

#assign taxonomy to each node representative with Silva 128 99 taxonomy. Max accepts is set to zero for
an exhaustive search.
assign_taxonomy.py -i Krill/2017/oligotyping/18S_node_representatives.fna -o
Krill/2017/oligotyping/taxonomy/ -t reference_sets/silva128/majority_taxonomy_7_levels.txt -r
reference_sets/silva128/99_otus_18S.fasta --uclust_max_accepts 0

#detect chimeras in final data set
identify_chimeric_seqs.py -i Krill/2017/oligotyping/18S_node_representatives.fna -o
Krill/2017/oligotyping/chimeric_otus.txt -t
reference_sets/silva128/majority_taxonomy_7_levels.txt -r
reference_sets/silva128/99_otus_18S.fasta -m blast_fragments -n 2 -d 4

#Remember to remove these chimeras from the OTU table, as well as removing any OTUs with a length
```

below 200 bp