Process MHC II DQB

Meinolf Ottensmann

Preface

All processing steps for analysing the Illumina MiSeq reads are outlined within this document. Most steps are based on several `python` scripts that may be executed using the command line (e.g. 'Bash on Ubuntu in Windows'), while `R` functions can be called within this document.

```
library(magrittr)
source("R/clustering_functions.R")
```

Prepare metafile

The first step of the processing uses the Qiime script `extract_barcodes.py` to reorient froward and reverse reads and trimming barcodes form the sequences. In order to do so, a metafile is needed that gives the forward and reverse primer sequences respectively which allow to recognise the orientation of a read.

```
## R
## read barcodes
barcodes <-
  read.csv("data/mhc_barcodes.csv", header = T, sep = ";", skip = 1)

## format left barcode
index_l <-
  barcodes[barcodes[["primer"]] == "Forward",c("Index","Sequence")]
names(index_l) <- c("index_L","BC_L")

## format right barcode
index_r <-
  barcodes[barcodes[["primer"]] == "Reverse",c("Index","Sequence")]
names(index_r) <- c("index_R","BC_R")

## read list of samples and join with barcodes
dqb <-
  read.csv("documents/dqb-samples-miseq.csv", header = T, sep = "\t") %>%
  dplyr::left_join(., index_l, by = "index_L" ) %>%
  dplyr::left_join(., index_r, by = "index_R")

## create metafile to erxtract barcodes using QIIME
dqb_qiime <- data.frame(
  SampleID =
    paste0(dqb[["sample"]],".",dqb[["position"]]),
  BarcodeSequence =
    paste0(dqb[["BC_L"]],dqb[["BC_R"]]),
  LinkerPrimerSequence =
    read.table("documents/mhc_primer_sequences.csv", header = T, sep = ",")[2,2] %>%
    as.character(),
  ReversePrimer =
    read.table("documents/mhc_primer_sequences.csv", header = T, sep = ",")[2,3] %>%
    as.character(),
```

```
  Description = NA)

dqb_qiime$SampleID <-
  stringr::str_replace(dqb_qiime$SampleID, "/",".")

## Manually add '#' before each sample in the created file
write.table(x = dqb_qiime,
            file = "miseq_reads/DQB-Pool/dqb_barcodes.txt",
            sep = "\t",
            row.names = F,
            quote = F)
```

Start processing MiSeq reads

```
# bash
#############################################################################################
# Prerequisites:                                                                            #
#                                                                                           #
# 'mhc_cluster' needs to be downloded from https://github.com/mottensmann/mhc_cluster       #
# and added to the 'PATH' variable.                                                         #
# Vsearch v.2.44, hmmer-3.1b2, usearch10, muscle3.8.31 are expected to be in 'PATH'         #
#                                                                                           #
# Folder that are missing in the PATH can be added using the following notation             #
# where '~' gives the path starting from the root.                                          #
# Example: export PATH="~/mhc_cluster:$PATH"                                                #
# Here, QIIME is assumed to be part of the conda environment                                #
# Raw reads may be downloaded as zip archieves and saved in a subfolder raw_reads           #
#############################################################################################

## Start conda
## -----------
source activate qiime191conda

## Set directory
## -------------
cd miseq_reads/DQB-Pool/

## MUSCLE alignment of dqb sequences
## ---------------------------------
muscle -in hmm/seal_dqb.fasta -out hmm/seal_dqb.afa ;

## create hidden markov model
## --------------------------
hmmbuild hmm/seal_dqb.hmm hmm/seal_dqb.afa ;

## create auxilariy files for hmmscan
## ----------------------------------
hmmpress hmm/seal_dqb.hmm ;

## copy to folder lib in repository mhc_cluster
## --------------------------------------------
cd hmm/
find -name "seal_dqb*" -print -exec cp {} ~/mhc_cluster/lib/ \;
cd ..

## Unzip raw reads
## ---------------
cd raw_reads
find . -iname "*.gz" -exec gunzip {} \;
cd ..

## Extract and strip barcodes from reads
## -------------------------------------
extract_barcodes.py --input_type barcode_paired_end
                    -f raw_reads/reads1.fastq -r raw_reads/reads2.fastq
```

```
                    --bc1_len 8 --bc2_len 8 -m dqb_barcodes.txt
                    -a -o parsed_barcodes/ ;


## Trim ends: Choosen length maximises the number of successfully merged reads
## ----------------------------------------------------------------------------
fastx_trimmer -i parsed_barcodes/reads1.fastq -o parsed_barcodes/reads1.fastq -l 239 &
fastx_trimmer -i parsed_barcodes/reads2.fastq -o parsed_barcodes/reads2.fastq -l 242 &

## Merge pairs of reads with vsearch
## ---------------------------------
vsearch --fastq_mergepairs parsed_barcodes/reads1.fastq
        --reverse parsed_barcodes/reads2.fastq
        --fastqout merged_reads/merged.fastq --fastq_maxdiffs 5
```

```
## R
## Filter merged reads
## -------------------
source("R/call_filter_merged_reads_dqb.R")
```

```
# bash
## Truncate filtered to get only exon sequences
## --------------------------------------------
fastx_trimmer -i merged_reads/merged_filtered.fastq
              -o merged_reads/merged_filtered_truncated.fastq -f 59 -l 240;

## Run MHC clustering pipeline
## ---------------------------
cluster_mhc2.py -f merged_reads/merged_filtered_truncated.fastq
                -o clustered_reads/dqb

## Explore sensitivity to parameter alpha
## --------------------------------------
cluster_mhc2.py -f merged_reads/merged_filtered_truncated.fastq
                -o clustered_reads/alpha_exploration/dqb
                 -alpha '0.0' ;
cluster_mhc2.py -f merged_reads/merged_filtered_truncated.fastq
                -o clustered_reads/alpha_exploration/dqb
                 -alpha '0.5' ;
cluster_mhc2.py -f merged_reads/merged_filtered_truncated.fastq
                -o clustered_reads/alpha_exploration/dqb
                 -alpha '1.0' ;
cluster_mhc2.py -f merged_reads/merged_filtered_truncated.fastq
                -o clustered_reads/alpha_exploration/dqb
                 -alpha '1.5' ;
cluster_mhc2.py -f merged_reads/merged_filtered_truncated.fastq
                -o clustered_reads/alpha_exploration/dqb
                 -alpha '2.0' ;
cluster_mhc2.py -f merged_reads/merged_filtered_truncated.fastq
                -o clustered_reads/alpha_exploration/dqb
                 -alpha '2.5' ;
cluster_mhc2.py -f merged_reads/merged_filtered_truncated.fastq
                -o clustered_reads/alpha_exploration/dqb
                 -alpha '3.0' ;
```

```
# R
## split reads in amplicons
demultiplex_fastq(
  reads = "miseq_reads/DQB-Pool/merged_reads/merged_filtered_truncated.fastq",
  out = "miseq_reads/DQB-Pool/merged_reads/demultiplexed",
  outname = "seq.fastq")
```

```
# bash
## Apply to samples individually
## -----------------------------
cd merged_reads/demultiplexed/
find . -type d | while read d; do
   (cd $d/
   cluster_mhc2.py -f seq.fastq -o dqb -alpha '0.0' ;
   cluster_mhc2.py -f seq.fastq -o dqb -alpha '0.5' ;
   cluster_mhc2.py -f seq.fastq -o dqb -alpha '1.0' ;
   cluster_mhc2.py -f seq.fastq -o dqb -alpha '1.5' ;
   cluster_mhc2.py -f seq.fastq -o dqb -alpha '2.0' ;
   cluster_mhc2.py -f seq.fastq -o dqb -alpha '2.5' ;
   cluster_mhc2.py -f seq.fastq -o dqb -alpha '3.0' ;
   )
done
cd ../
```

```
## R
## Remove temporary created files
remove_mhc_cluster_files(
  parentfolder = "miseq_reads/DQB-Pool/merged_reads/demultiplexed/",
  fastq = "seq.fastq")

## Pool alleles of all individually processed amplicons
filen <- c("dqb_pct_1.0_a_0.0_ee_1.0.fixed.otus.fa",
           "dqb_pct_1.0_a_0.5_ee_1.0.fixed.otus.fa",
           "dqb_pct_1.0_a_1.0_ee_1.0.fixed.otus.fa",
           "dqb_pct_1.0_a_1.5_ee_1.0.fixed.otus.fa",
           "dqb_pct_1.0_a_2.0_ee_1.0.fixed.otus.fa",
           "dqb_pct_1.0_a_2.5_ee_1.0.fixed.otus.fa",
           "dqb_pct_1.0_a_3.0_ee_1.0.fixed.otus.fa")
lapply(filen, function(x) {
  pool_zotus(parentfolder = "miseq_reads/DQB-Pool/merged_reads/demultiplexed/",
             filen = x)})
```

```
# bash
## Cluster using based on individually identified Zotus
## ----------------------------------------------------
cluster_against_db.py -f merged_reads/merged_filtered_truncated.fastq
   -ref merged_reads/demultiplexed/dqb_pct_1.0_a_0.0_ee_1.0.fixed.otus.fa_pooled_zotus.fa
   -o reference_based/a_0.0 ;
cluster_against_db.py -f merged_reads/merged_filtered_truncated.fastq
   -ref merged_reads/demultiplexed/dqb_pct_1.0_a_0.5_ee_1.0.fixed.otus.fa_pooled_zotus.fa
   -o reference_based/a_0.5 ;
cluster_against_db.py -f merged_reads/merged_filtered_truncated.fastq
   -ref merged_reads/demultiplexed/dqb_pct_1.0_a_1.0_ee_1.0.fixed.otus.fa_pooled_zotus.fa
   -o reference_based/a_1.0 ;
cluster_against_db.py -f merged_reads/merged_filtered_truncated.fastq
```

```
   -ref merged_reads/demultiplexed/dqb_pct_1.0_a_1.5_ee_1.0.fixed.otus.fa_pooled_zotus.fa
   -o reference_based/a_1.5  ;
cluster_against_db.py -f merged_reads/merged_filtered_truncated.fastq
   -ref merged_reads/demultiplexed/dqb_pct_1.0_a_2.0_ee_1.0.fixed.otus.fa_pooled_zotus.fa
   -o reference_based/a_2.0 ;
cluster_against_db.py -f merged_reads/merged_filtered_truncated.fastq
   -ref merged_reads/demultiplexed/dqb_pct_1.0_a_2.5_ee_1.0.fixed.otus.fa_pooled_zotus.fa
   -o reference_based/a_2.5 ;
cluster_against_db.py -f merged_reads/merged_filtered_truncated.fastq
   -ref merged_reads/demultiplexed/dqb_pct_1.0_a_3.0_ee_1.0.fixed.otus.fa_pooled_zotus.fa
   -o reference_based/a_3.0 ;
```

```
sessionInfo()
> R version 3.4.3 (2017-11-30)
> Platform: x86_64-w64-mingw32/x64 (64-bit)
> Running under: Windows 10 x64 (build 16299)
>
> Matrix products: default
>
> locale:
> [1] LC_COLLATE=English_United Kingdom.1252
> [2] LC_CTYPE=English_United Kingdom.1252
> [3] LC_MONETARY=English_United Kingdom.1252
> [4] LC_NUMERIC=C
> [5] LC_TIME=English_United Kingdom.1252
>
> attached base packages:
> [1] stats     graphics  grDevices utils     datasets  methods
> [7] base
>
> other attached packages:
> [1] magrittr_1.5 knitr_1.17
>
> loaded via a namespace (and not attached):
>  [1] compiler_3.4.3  backports_1.1.2 rprojroot_1.3-1 tools_3.4.3
>  [5] htmltools_0.3.6 yaml_2.1.16     Rcpp_0.12.14    stringi_1.1.6
>  [9] rmarkdown_1.8   stringr_1.2.0   digest_0.6.13   evaluate_0.10.1
```