This tutorial demonstrates how to generate shell scripts for running PLEKModelling in parellel.

1. Pre-processing steps:

(1) Install PLEK under this folder:

/dat2/plek/demo/PLEK.1.2/

(2) Copy mRNA and lncRNA transcript file (used to train model) to the tutorial foder:

cd /dat2/plek/demo/PLEK.1.2/

mkdir tutorial

cp PLEK_mRNAs.fa ./tutorial/

cp PLEK_lncRNAs.fa ./tutorial/

(3) Copy test data file (used to test the model/classify transcripts) to the tutorial foder:

cp PLEK_test.fa ./tutorial/

(4) Copy *PLEK_generate_scripts.R* (used to test the model/classify transcripts) to the tutorial foder:

cp *PLEK_generate_scripts.R* ./tutorial/

cd tutorial

ls

2.Configure PLEK_generate_scripts.R script.

vim *PLEK_generate_scripts.R*

(1) *PLEK_generate_scripts.R* is run in:

/dat2/plek/demo/PLEK.1.2/tutorial/

(2) Shell scripts will be run in:

/dat2/plek/demo/PLEK.1.2/tutorial/

(3) PLEKModelling.py is located in:

/dat2/plek/demo/PLEK.1.2/

(4) PBS queue name (-q) is:

batch

(5) mRNA transcript file is:

/dat2/plek/demo/PLEK.1.2/tutorial/PLEK_mRNAs.fa

(6) lncRNA transcript file is:

/dat2/plek/demo/PLEK.1.2/tutorial/PLEK_lncRNAs.fa

(7) The files produced by shell scripts will have this prefix:

myprefix

(8) Job name that will be displayed on PBS/Torque:

pbsjob

(9) n-fold cross-validation:

'2' for demo; '10' is recommended.

(10) *C* and *Gamma* parameters for training model.

log2c_from=0

log2c_to=5

log2c_by=1

log2g_from=0

log2g_to=-5

log2g_by=-1

(11) K-mers

'k=4' for demo; 'k=5' is recommended.

(12) Save file

3.Build a new model:

(1) Start R and run "PLEK_generate_scripts.R".

source("PLEK_generate_scripts.R")

(2) Exit R.

q()

(3) This R script will generate 36 Shell script files.

ls /dat2/plek/demo/PLEK.1.2/tutorial/*.sh

(4) Pick up one to test if it can run, CTRL+C to terminate.

more /dat2/plek/demo/PLEK.1.2/tutorial/myprefix_p01.sh

(5) To run shell scripts (This step is time-consuming.

It depends on number of samples, n-fold, k-mers):

chmod u+x /dat2/plek/demo/PLEK.1.2/tutorial/myprefix_qsub_file_list.txt

. /dat2/plek/demo/PLEK.1.2/tutorial/myprefix_qsub_file_list.txt

qstat

(6) Please check the *modelling.logs files.

These files log the run process of PLEKModelling.py

ls /dat2/plek/demo/PLEK.1.2/tutorial/*modelling.logs

more /dat2/plek/demo/PLEK.1.2/tutorial/myprefix_p01_modelling.logs

(7) Check error logs:

ll *.err

(8) View all the parameters and results.

Find the model with the highest rate.

grep "rate=" /dat2/plek/demo/PLEK.1.2/tutorial/*.sh.out

Get the best rate (the last line):

grep "rate=" /dat2/plek/demo/PLEK.1.2/tutorial/*.sh.out | sed 's/out:.*rate=/out\t/g' | sed 's/).*//g' | sort -n -k 2

The best model is:

/dat2/plek/demo/PLEK.1.2/tutorial/myprefix_p??.model

/dat2/plek/demo/PLEK.1.2/tutorial/myprefix_p??.range

4.Test the new model:

```
python  /dat2/plek/demo/PLEK.1.2/PLEK.py  \
  -fasta  /dat2/plek/demo/PLEK.1.2/tutorial/PLEK_test.fa  \
  -out  /dat2/plek/demo/PLEK.1.2/tutorial/PLEK_test.predicted \
  -thread  10  \
  -range  /dat2/plek/demo/PLEK.1.2/tutorial/myprefix_p??.range  \
  -model  /dat2/plek/demo/PLEK.1.2/tutorial/myprefix_p??.model  \
  -k 4
```

Note: ‘k=4’ for demo; ‘k=5’ is recommended.