Running `bitseq`

Panagiotis Papastamoulis

University of Manchester

The following example first uses `bowtie` to align a set of reads to a reference transcriptome. We then run `bitseq` in order to estimate transcript expression (`estimateExpression`). Finally, `estimateDE` is used in order to detect differentially expressed transcripts between two conditions. The following modules should be loaded:

```
module load apps/gcc/samtools/0.1.18
module load apps/binapps/bowtie2/2.1.0
module load apps/gcc/bitseq/0.7.0
```

0.1 Reference annotation and mapping with `bowtie`

Download a reference annotation from UCSC Table Browser:

`http://genome.ucsc.edu/cgi-bin/hgTables`

Denote by `referenceTranscriptome` the full directory path containing the fasta (`ref.fa`) file with the reference transcriptome. Full details for selecting the right configuration of UCSC tables are given in the following link:

`http://code.google.com/p/bitseq/wiki/BitSeq`

Assume that our sample consists of two biological conditions (`A` and `B`) and each one consists of two sets of reads (`.fastq` files):

```
conditionA_1.fastq conditionA_2.fastq
conditionB_1.fastq conditionB_2.fastq
```

The following jobscript will map the four available samples to the reference annotation using `bowtie`.

```
## SGE Stuff
### Use the current/submission directory as the working directory
#$ -cwd
### Inherit the user environment settings from the login node
#$ -V
#$ -pe smp.pe 4 #### Run in parallel with 4 threads
export OMP_NUM_THREADS=$NSLOTS

# build the bowtie2 index files
cd referenceTranscriptome
bowtie2-build -f ref.fa ref
# Now change to your working directory (WD) containing the .fastq files
cd WD
```

```
# map the reads for all samples:
bowtie2 -q -k 50 -p 4 referenceTranscriptome/ref conditionA_1.fastq -S A_1.sam
bowtie2 -q -k 50 -p 4 referenceTranscriptome/ref conditionA_1.fastq -S A_2.sam
bowtie2 -q -k 50 -p 4 referenceTranscriptome/ref conditionA_1.fastq -S B_1.sam
bowtie2 -q -k 50 -p 4 referenceTranscriptome/ref conditionA_1.fastq -S B_2.sam
```

Note that the `sam` files are containing your mapped reads.

0.2 Running `bitseq`

Now we can estimate expression and perform DE analysis using `bitseq`. At first we will pre-process our reads using the `parseAlignment` command by assuming a uniform read distribution. Next we will estimate transcript expression (`estimateExpression`) and finally we will discover differentially expressed transcripts (`estimateDE`).

```
## SGE Stuff
### Use the current/submission directory as the working directory
#$ -cwd
### Inherit the user environment settings from the login node
#$ -V
#$ -pe smp.pe 4 #### Run in parallel with 4 threads
export OMP_NUM_THREADS=$NSLOTS

#pre-processing:
parseAlignment  A_1.sam -o A_1.prob --trSeqFile referenceTranscriptome/ref.fa \
--trInfoFile data.tr --uniform --verbose
# for subsequent runs trInfoFile is not needed
parseAlignment  A_2.sam -o A_2.prob --trSeqFile referenceTranscriptome/ref.fa \
 --uniform --verbose
parseAlignment  B_1.sam -o B_1.prob --trSeqFile referenceTranscriptome/ref.fa \
 --uniform --verbose
parseAlignment  B_2.sam -o B_2.prob --trSeqFile referenceTranscriptome/ref.fa \
 --uniform --verbose

#estimate expression for sample A1. Output written to A_1.rpkm
estimateExpression A_1.prob -o A_1 --outType RPKM -p A_1.txt -t data.tr -P 4
#estimate expression for sample A2. Output written to A_2.rpkm
estimateExpression A_2.prob -o A_2 --outType RPKM -p A_2.txt -t data.tr -P 4
#estimate expression for sample B1. Output written to B_1.rpkm
estimateExpression B_1.prob -o B_1 --outType RPKM -p B_1.txt -t data.tr -P 4
#estimate expression for sample B2. Output written to B_2.rpkm
estimateExpression B_2.prob -o B_2 --outType RPKM -p B_2.txt -t data.tr -P 4

# Perform Differential Expression analysis between the two conditions:
getVariance --log -o data.Lmean dataA_1.rpkm dataA_2.rpkm dataB_1.rpkm dataB_2.rpkm
estimateHyperPar --meanFile data.Lmean -o data.param \
                   dataA_1.rpkm dataA_2.rpkm C dataB_1.rpkm dataB_2.rpkm
estimateDE -o data -p data.param dataA_1.rpkm dataA_2.rpkm C dataB_1.rpkm dataB_2.rpkm
```

Note that the DE analysis results are saved to the `data.pplr` file.