# Usage Information

**MISO**

**Run MISO:**

1) Index the annotation using index\_gff:

index\_gff --index refGene.gff3 indexed/

where refGene.gff3 is a GFF file containing descriptions of isoforms/alternative splicing events to be quantitated (e.g. skipped exons).

2) Run MISO

miso --run indexed/ sample1.bam --output-dir output1/ --read-len 76 –use-cluster

miso --run indexed/ sample2.bam --output-dir output2/ --read-len 76 –use-cluster

The --read-len option is necessary and specifies the length of the reads in the data

To compute expression levels using paired-end reads, use the --paired-end option

3) Summarize MISO inferences using summarize\_miso --summarize-samples:

summarize\_miso --summarize-samples output1/ summary\_output1/

summarize\_miso --summarize-samples output2/ summary\_output2/

4) Make pairwise comparisons between samples to detect differentially expressed isoforms/events with compare\_miso --compare-samples:

compare\_miso --compare-samples output1/ output2/ comparison\_output/

**Roar**

**Run roar:**

library(roar)

gtf <- system.file("examples", "apa.gtf", package="roar")

bamTreatment <- c("treatment1.bam","treatment2.bam","treatment3.bam")

bamControl <- c("control1.bam","control2.bam")

rds <- RoarDatasetFromFiles(bamTreatment, bamControl, gtf)

1)It counts reads overlapping with the PRE/POST portions defined in the given gtf/GRanges annotation。

rds <- countPrePost(rds, FALSE)

2)It computes the ratio of prevalence of the short and long isoforms for every gene in the treatment and control condition (m/M) and their ratio, roar, that indicates if there is a relative shortening-lengthening in a condition over the other one.

rds <- computeRoars(rds)

3)It applies a Fisher test comparing counts falling on the PRE and POST portion in the treatment and control conditions for every gene.

rds <- computePvals(rds)

4)It returns a dataframe containing m/M values, roar values and pvalues

results <- totalResults(rds)

results\_filtered <- pvalueFilter(rds, fpkmCutoff=-Inf, pvalCutoff=0.05)

**QAPA**

**Run QAPA:**

QAPA has three sub-commands:

build: Generate a 3’UTR library from annotations

fasta: Extract sequences for indexing by transcript quantification tools

quant: Calculate relative usage of alternative 3’UTR isoforms

1) To extract 3′ UTRs from annotation

qapa build --db ensembl\_identifiers.txt -g gencode.polyA\_sites.bed -p clusters.mm10.bed gencode.basic.txt > output\_utrs.bed

ensembl\_identifiers.txt and gencode.basic.txt are gene annotation files

gencode.polyA\_sites.bed and clusters.mm10.bed are the ploy(A) annotation files

**If using a custom BED file, replace the -g and -p options with -o:**

qapa build --db ensembl\_identifiers.txt -o custom\_sites.bed gencode.basic.txt > output\_utrs.bed

2) To extract sequences from the BED file prepared by build, a reference genome in FASTA format is required.

qapa fasta -f genome.fa output\_utrs.bed output\_sequences.fa

Note that genome.fa must be uncompressed

3) Expression quantification of 3’UTR isoforms must be carried out first using the FASTA file prepared by fasta as the index.

To index the sequences using Salmon: salmon index -t output\_sequences.fa -i utr\_library

qapa quant --db ensembl\_identifiers.txt project/sample\*/quant.sf > pau\_results.txt

**PAQR**

**Run PAQR:**

Configure the input parameters:

The PAQR subdirectory contains a file called "config.yaml". This files contains all information about used parameter values, data locations, file names and so on. During a run, all steps of the pipeline will retrieve their parameter values from this file.

1、max\_cores=4 # maximum number of threads that will run in parallel

snakemake -s part\_one.Snakefile -p --cores ${max\_cores} &> log\_output.log

2、max\_cores=8 # maximum number of threads that will run in parallel

snakemake -s part\_two.Snakefile -p --cores ${max\_cores} &>> log\_output.log

**Also the single steps/scripts of the pipeline**

1) python calculate-TIN-values.py \

-i data/bam\_files/KD\_rep1.bam \

-r ${transcripts} \

-c ${min\_raw\_reads} \

--names KD\_rep1 \

-n ${sample\_size} \

> data/samplewise\_TIN/KD\_rep1.tsv

2) python merge-TIN-tables.py \

--verbose \

--input data/samplewise\_TIN/KD\_rep1.tsv data/samplewise\_TIN/CTL\_rep1.tsv \

> HNRNPC\_KD/bias.transcript\_wide.TIN.tsv

3) Rscript boxplots-TIN-distributions.R \

--file HNRNPC\_KD/bias.TIN.median\_per\_sample.tsv\

--pdf HNRNPC\_KD/bias.transcript\_wide.TIN.boxplots.pdf

**Run KAPAC:**

Rscript --vanilla KAPAC.R --help

**Cufflinks**

**Run Cufflinks:**

cufflinks -p 8 -G transcript.gtf -o OUTPUT\_FILE BAM\_FILE

**ExUTR**

**Run ExUTR:**

1) Reading Open Frame (ORF) prediction

perl 3UTR\_orf.pl -i transcripts.fasta -d /home/user/swissprot/swissprot -a 8 -o Test -l un

2) 3'UTR sequence retrieval

perl 3UTR\_ext.pl -i1 Test\_transcripts.fa -i2 Test\_orfs.fasta -d /home/user/3UTR\_database/3UTR.mam.fasta -a 8 -o 3UTR.fasta -x 2500 -m 20

**Scripture**

**Run Scripture:**

1) Make paired end alignment files using Scripture

Remove the headers from the TopHat alignment file (headers begin with "@") and sort each by read name:

>> sed '1,2d' accepted\_hits1.sam | sort > accepted\_hits1.sorted.sam

>> sed '1,2d' accepted\_hits2.sam | sort > accepted\_hits2.sorted.sam

Run Scripture with task makePairedFile:

>> java -Xmx4000m -jar scripture.jar -task makePairedFile -pair1 accepted\_hits1.sorted.sam -pair2 accepted\_hits2.sorted.sam -out XXXXX.paired.sam –sorted

2) Run Scripture

Combine TopHat alignment file, then sort and index:

>> cat accepted\_hits1.sorted.sam accepted\_hits.sorted2.sam > all\_alignments.sam

Combine paired end alignments, then sort and index

>> cat XXXXX1.paired.sam XXXXX2.paired.sam XXXXX3.paired.sam > all\_alignments.paired.sam

>>java –jar scripture.jar –alignment all\_alignments.sorted.sam –out  scriptureESTest –sizeFile hg19.sizes

–chr chr19 –chrSequence chr19.fa -pairedEnd all\_alignments.paired.sorted.sam

**KLEAT**

**Run KLEAT:**

1) Use the included script (TA.sh) to generate the input necessary for KLEAT

./TA.sh -a <in1.fq.gz> -b <in2.fq.gz> -n <sample name> -k <space-delimited k-values>

-o/output/path/for/assembly -t <number threads> -m <memory for sorting bam>

Example:

./TA.sh -a sample\_1.fq.gz -b sample\_2.fq.gz -n DHX30 -k "32 52 72" -o test/assembly -t 6 -m 15G

2) Run KLEAT :

python KLEAT.py <assembly\_dir>/c2g.bam <assembly\_dir>/sample-merged.fa <reference\_genome.fa> <gtf> <assembly\_dir>/r2c\_sorted.bam /output/path/prefix -k <track\_name> <track\_description> -x –ss

Example:

python KLEAT.py /assembly/c2g.bam /assembly/merged/DHX30-merged.fa hg19.fa ensembl.fixed.sorted.gz /assembly/r2c\_sorted.bam /KLEAT/DHX30 -k KLEAT\_test "KLEAT cleavage sites" -ss

**ContextMap2**

**Run ContextMap2:**

java -jar ContextMap\_v2.1.0.jar mapper <main arguments> [options]\*

**Main arguments:**

-reads: A comma separated list of file paths to reads in fasta/fastq/zip/gz format. A single path for single-end mapping and two paths (#1 mates and #2 mates) for paired-end mapping

-aligner\_name: The following alignment tools are supported: "bwa", "bowtie1" or "bowtie2".

-aligner\_bin: The absolute path to the executable of the chosen aligner tool.

-indexer\_bin: The absolute path to the executable of the aligner's indexing tool (not needed for BWA).

-indices: A comma separated list of paths to basenames of indices, which can be used by the chosen aligner.

-genome: The path to a folder with reference sequences in fasta format (for each chromosome a separate file).

-o: The path to the output directory.

**Option:**

--ployA: Enables the search for polyA-tails. This option is mutually exclusive with the --noclipping option. (default: off)

**GETUTR**

**Run GETUTR：**

python GETUTR.py -i xxx.bam -o xxx.3UTR -m 10 -r refFlat.txt

-i <inputfile name> RNA-seq mapped reads in BAM or BED format.

-o <outputfile name> inputfile name without extension

the name-head of output files

<outputfile name> smoothed.bed: estimated 3' UTR

<outputfile name> PCS: poly(A) cleavage site

-m <index of method> Default:10 (PAVA) 10: PAVA / 0: Max.fit / 1: Min.fit.

-r <reference file> reference file in GenePred/GTF format for gene annotation

**PHMM**

**Run PHMM:**

1) build transcript database

Rscript buildTranscriptDB\_fixed.R refGene txdb.hg19.refGene.sqlite hg19

2) txdb.hg19.refGene.sqlite select transcripts with 3'utr with length > 600bp

Rscript apa3utr\_fixed.R txdb.hg19.refGene.sqlite 22(number of chr) long3utr.txt

3、bam file long3utr.txt compute read tag counts in sliding windows

Rscript apaCount\_fixed.R long3utr.txt 22 myfile.sorted.bam

4、long3utr.txt myfile.sorted.cts.rda fit poisson HMM

Rscript poissonHMM\_fixed long3utr.txt myfile.sorted.cts.rda res.myfile.csv

**ChangePoint**

**Run ChangePoint:**

perl change\_point.pl -t <treatment.bam> -c <control.bam> -g <utr.bed> -d <s/l> [options]

-d: s for detecting shortening events or l for detecting lengthening events

**Options:**

-o STRING The prefix for output file. The default value is change\_point.

-n INT At least how many reads support each region. The default is 20. In either condition, if any region has reads smaller than this threshold, the region will be discarded.

-a FLOAT Mixed directional FDR level. The default is 0.05.

-r INT The hypothesized odds ratio in Fisher's Exact Test. The default is 2, meaning that the program will tend to select significant changes with odds ratio greater than 2.

-s STRING The maximum heap size for JVM. The default is 256m.

-x STRING The minimum heap size for JVM. The default is 1024m.

**IsoSCM**

**Run IsoSCM:**

1) Assemble command

java -Xmx2048m -jar IsoSCM.jar assemble -bam brain.bam -base brain -s reverse\_forward

**Required arguments:**

-bam BAM file to be assembled

-base The output basename. This will be used as the prefix for output files.

-s the strandedness, can be "reverse\_forward" or "unstranded"

**Assembly arguments:**

-coverage whether or not to calculate coverage of generated models. Default: true

-dir The output directory. Default: ./isoscm

-insert\_size\_quantile If the data is paired, and this value is specified IsoSCM will attempt to scaffold gaps between segments spanned by mates separated by at most this quantile

-internal whether or not to identify change-points within internal (non-terminal) exons.

Default: false

-jnct\_alpha The significance level for binomial test to accept splice junction.

Default: 0.05

-merge\_radius gaps smaller than this distance (nt) will be merged. Default: 100

-merge\_segments Optional: bed file of regions to merge across

**Segmentation arguments:**

-min\_fold the minimum fold change between neighboring segments expressed as a ratio of coverage downstream divided by coverage upstream. acceptable values in the range [0.0-1.0]. Default: 0.5

-min\_terminal terminal segments are extended bythis amount before segmentation. Default: 300

-nb\_r r parameter of the NB(p,r) prior on RNA-seq coverage within a segment. The p parameter is not specified as it is integrated over. Default: 10

-segment\_p p parameter of NB(p,r) prior on segment length (between change-points). Default: 0.95

-segment\_r r parameter of NB(p,r) prior on segment length (between change-points). Default: 10

-t the threshold above which a segment is considered expressed. Default: 1

-w the width of window to be used by the SCM. Default: 20

2、Compare command

java -Xmx2048m -jar IsoSCM.jar compare [arguments]

Required arguments:

-base The output basename. This will be used as the prefix for output files.

-x1 XML configuration file for the assembly step from sample 1

-x2 XML configuration file for the assembly step from sample 2

Optional arguments:

-dir The output directory. Default: ./compare

**DaPars**

**Run DaPars:**

1) Generate region annotation

python DaPars\_Extract\_Anno.py -b gene.bed -s symbol\_map.txt -o extracted\_3UTR.bed

-b bed file：The gene model in BED format

-s symbol\_map.txt：The mapping of transcripts to gene symbol

2) python DaPars\_main.py configure\_file

configure\_file:

#The following file is the result of step 1.

Annotated\_3UTR=hg19\_refseq\_extracted\_3UTR.bed

#A comma-separated list of BedGraph files of samples from condition 1

Group1\_Tophat\_aligned\_Wig=Condition\_A\_chrX.wig

#A comma-separated list of BedGraph files of samples from condition 2

Group2\_Tophat\_aligned\_Wig=Condition\_B\_chrX.wig

Output\_directory=DaPars\_Test\_data/

Output\_result\_file=DaPars\_Test\_data

#At least how many samples passing the coverage threshold in two conditions

Num\_least\_in\_group1=1

Num\_least\_in\_group2=1

Coverage\_cutoff=30

#Cutoff for FDR of P-values from Fisher exact test.

FDR\_cutoff=0.05

PDUI\_cutoff=0.5

Fold\_change\_cutoff=0.59

**APAtrap**

**Run APAtrap:**

1. identifyDistal3UTR

1) For genome having long 3'UTR,

identifyDistal3UTR -i Sample1.bedgraph Sample2.bedgraph -m hg19.genemodel.bed -o novel.utr.bed

2) For genome having short 3'UTR,

identifyDistal3UTR -i Sample1.bedgraph Sample2.bedgraph -m rice.genemodel.bed -o novel.utr.bed -w 50 -e 5000

**Necessary parameters:**

-i short reads mapping result in bedgraph/wig format, can accept single file or multiple files.

-m gene model file in bed format.

-o file store the information of extended 3'UTR in bed format.

**Optional parameters:**

-w window size used to scan the mapping result, default is 100.

-e pre-extension size of each 3'UTR, default is 10000.

-c minimum coverage of the end of the distal 3'UTR with comparing to the whole transcript.

-p minimum percentage of valid nucleotides in a scanning-window.

-s gene symbol file.

2. predictAPA

1) For genome having long 3'UTR,

predictAPA -i Sample1.bedgraph Sample2.bedgraph -g 2 -n 1 1 -u hg19.utr.bed -o output.txt

2) For genome having short 3'UTR,

predictAPA -i Sample1.bedgraph Sample2.bedgraph -g 2 -n 1 1 -u rice.utr.bed -o output.txt -a 50

**Necessary parameters:**

-i short reads mapping result in bedgraph/wig format, can accept single file or multiple files.

-g number of groups (treatments/conditions) of the input files, e.g. -g 2.

-n number of files(biological replicates) in each group (treatment/condition), e.g. -n 1 1.

-u 3'UTR annotation file in bed format.

-o information of the predicted APA sites and their usage.

**Optional parameters:**

-d minimum degree of coverage variation between two adjcent APA sites, >0 and <1,

default is 0.2.

-c minimum average coverage required for each 3'UTR, >=10, default is 20.

-a minimum distance between the predicted APA sites, >=20, default is 100.

-w window size used to scan the profile, >=20, default is 50.

3. deAPA(R)

library(deAPA)

deAPA(input\_file, output\_file, group1, group2, least\_qualified\_num\_in\_group1, least\_qualified\_num\_in\_group2, coverage\_cutoff)

**Arguments:**

input\_file: The result generated by 'predictAPA'.

output\_file: Name of output file.

group1: The first group of sample to be compared, default is 1.

group2: The second group of sample to be compared, default is 2.

least\_qualified\_num\_in\_group1: Minimum number of qualified replicates in sample group1, default is 1.

least\_qualified\_num\_in\_group2: Minimum number of qualified replicates in sample group2, default is 1.

coverage\_cutoff: Minimum coverage depth required for each sample, default is 20.

**TAPAS**

**Run TAPAS:**

1) APA site detection

./APA\_sites\_detection -ref <annotation> -cov <A read coverage file> -l < Read length > -o <output file>

2) Differential APA site analysis:

./Diff\_APA\_site\_analysis -C1 cond1\_r1.txt,cond1\_r2.txt,cond1\_r3.txt,cond1\_r4.txt -C2 cond2\_r1.txt,cond2\_r2.txt,cond2\_r3.txt,cond2\_r4.txt -a refFlat\_sf.txt -cutoff 70 -type s/d -o deci\_output.txt

**OPTIONS:**

-C1 <cond1\_f1,cond1\_f2,cond1\_f3,..> Comma separated file names of condition 1 are given using this option. Each of these filesis the APA site detection file, outputted by the first part of TAPAS (outputted by APA\_sites\_detection).

-C2 <cond2\_f1,cond2\_f2,cond2\_f3,..> Comma separated file names of condition 2 are given using this option. Each of these files is the APA site detection file, outputted by the first part of TAPAS (outputted by APA\_sites\_detection).

-a <annotation\_file\_name> An annotation file is given using this option. This file is similar to the annotation fileof the APA site detection analysis.

-cutoff <int> Cutoff value is given using this option. This parameter is explained in TAPAS manuscript. Default value: 70

-type <d/s> Type of differential analysis.

d-> differential APA site analysis, s -> shortening/lengthening event analysis.

-o <output\_file\_name> Ouput file name is given using this option.

**EBChangePoint**

**Run EBChangePoint:**

perl   EBChangePoint.pl   -c   control.bam   -t   case.bam   -g   exon.bed

-h1   junctions\_control.bed   -h2   junctions\_case.bed

**Required:**

-c FILE Control sample in bam format.

-t FILE Case sample in bam format.

-g FILE Regions to be analyzed in bed format, such as exon regions.

-h1 FILE Junction.bed file for control sample, i.e. generated by Tophat.

-h2 FILE Junction.bed file for case sample, i.e. generated by Tophat.

**Options:**

-o STRING Output file. [change\_point]

-a FLOAT Mixed directional FDR level. [0.05]

-s STRING Max heap size for JVM. [256m]

-x STRING Min heap size for JVM. [1024m]

Parameters Influence

**APAtrap**

predictAPA:

predictAPA.pl -i syn10001.PolyA.sorted.bedgraph syn1002.noPolyA.sorted.bedgraph -g 2 -n 1 1 -u hg19.utr.bed -o output.txt -d 0.1 –a 100 –c 20

predictAPA.pl -i syn10001.PolyA.sorted.bedgraph syn1002.noPolyA.sorted.bedgraph -g 2 -n 1 1 -u hg19.utr.bed -o output.txt -d 0.1 –a 100 –c 30

predictAPA.pl -i syn10001.PolyA.sorted.bedgraph syn1002.noPolyA.sorted.bedgraph -g 2 -n 1 1 -u hg19.utr.bed -o output.txt -d 0.1 –a 100 –c 50

predictAPA.pl -i syn10001.PolyA.sorted.bedgraph syn1002.noPolyA.sorted.bedgraph -g 2 -n 1 1 -u hg19.utr.bed -o output.txt -d 0.1 –a 50 –c 20

predictAPA.pl -i syn10001.PolyA.sorted.bedgraph syn1002.noPolyA.sorted.bedgraph -g 2 -n 1 1 -u hg19.utr.bed -o output.txt -d 0.1 –a 150 –c 20

**DaPars**

python DaPars\_main.py configure\_file

configure\_file:

Annotated\_3UTR=hg19\_refseq\_extracted\_3UTR.bed

Group1\_Tophat\_aligned\_Wig=Condition\_A\_chrX.wig

Group2\_Tophat\_aligned\_Wig=Condition\_B\_chrX.wig

Output\_directory=DaPars\_Test\_data/

Output\_result\_file=DaPars\_Test\_data

Num\_least\_in\_group1=1

Num\_least\_in\_group2=1

Coverage\_cutoff=30

FDR\_cutoff=0.05

PDUI\_cutoff=0.5

Fold\_change\_cutoff=0.59

We change the Coverage\_cutoff and PDUI\_cutoff parameters:

PDUI\_cutoff=0.5:

Coverage\_cutoff=10; Coverage\_cutoff=20; Coverage\_cutoff=30;

Coverage\_cutoff=30:

PDUI\_cutoff=0.1; PDUI\_cutoff=0.3; PDUI\_cutoff=0.5; PDUI\_cutoff=0.7

**ChangePoint**

perl change\_point.pl -t treatment.bam -c control.bam -g utr.bed -d <s/l> -n 20 –a 0.05

perl change\_point.pl -t treatment.bam -c control.bam -g utr.bed -d <s/l> -n 20 –a 0.1

perl change\_point.pl -t treatment.bam -c control.bam -g utr.bed -d <s/l> -n 20 –a 0.3

perl change\_point.pl -t treatment.bam -c control.bam -g utr.bed -d <s/l> -n 20 –a 0.5

perl change\_point.pl -t treatment.bam -c control.bam -g utr.bed -d <s/l> -n 5 –a 0.05

perl change\_point.pl -t treatment.bam -c control.bam -g utr.bed -d <s/l> -n 10 –a 0.05

perl change\_point.pl -t treatment.bam -c control.bam -g utr.bed -d <s/l> -n 15 –a 0.05

perl change\_point.pl -t treatment.bam -c control.bam -g utr.bed -d <s/l> -n 30 –a 0.05

perl change\_point.pl -t treatment.bam -c control.bam -g utr.bed -d <s/l> -n 50 –a 0.05

**IsoSCM**

IsoSCM has many parameters, we select the two main parameters:

-min terminal (terminal segments are extended by this amount before segmentation);

-min fold (the minimum fold change between neighboring segments expressed as a ratio of coverage downstream divided bycoverage upstream. acceptable values in the range [0.0-1.0])

java -Xmx102400m -jar IsoSCM-2.0.11.jar assemble -coverage false -bam sample.bam -base output\_name -s unstranded -min terminal 300 -min fold 0.5 -jnct alpha 0.05

java -Xmx102400m -jar IsoSCM-2.0.11.jar assemble -coverage false -bam sample.bam -base output\_name -s unstranded -min terminal 50 -min fold 0.08 -jnct alpha 0.05

**GETUTR**

python GETUTR.py -i sample.bam -o output.3UTR -m 10 -r refFlat.txt

python GETUTR.py -i sample.bam -o output.3UTR -m 1 -r refFlat.txt

python GETUTR.py -i sample.bam -o output.3UTR -m 0 -r refFlat.txt