Contents

[**References** 2](#_Toc171411190)

[**STAR** 3](#_Toc171411191)

[**RSEM** 4](#_Toc171411192)

[**RNA-seq metrics** 5](#_Toc171411193)

[**SparK** 7](#_Toc171411194)

[**ARTDeco** 8](#_Toc171411195)

[**IsoformSwitchAnalyzeR External Sequence Analysis** 9](#_Toc171411196)

**References**

**EBV gene annotation inversion and liftover**

# Liftoff version: 1.6.3

# PMID: 33320174

# Separated the lytic LMP1 isoform into its own gene utilizing the NC\_007605 annotation from PMID: 34705568.

# Inverted NC\_007605 and M81 genomes to align their origins with the inverted Akata genome (PMID: 27873270).

# NC\_007605 new origin: 104,401

# M81 new origin: 108,346

liftoff -g NC\_007605.chrEBV.gtf -o NC\_007605.chrEBV.inverted.gtf -dir . NC\_007605.chrEBV.inverted.fa NC\_007605.chrEBV.fa

liftoff -g M81\_DFLR.chrEBV.gtf -o M81\_DFLR.chrEBV.inverted.gtf -dir . M81\_DFLR.chrEBV.inverted.fa M81\_DFLR.chrEBV.fa

# Performed liftover using the inverted NC\_007605 references and the inverted M81 genome.

liftoff -g NC\_007605.chrEBV.inverted.gtf -o M81\_DFLR.chrEBV.inverted.gtf -dir . M81\_DFLR.chrEBV.inverted.fa NC\_007605.chrEBV.inverted.fa

**Human genome reference**

# Download human reference files

# GENCODE primary assembly v45

wget https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode\_human/release\_45/gencode.v45.primary\_assembly.annotation.gtf.gz

# GENCODE genome sequence (GRCh38.p14)

wget <https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_45/GRCh38.p14.genome.fa.gz>

**Concatenate human, EBV, and ERCC references**

# Concatenate human, ERCC, and EBV annotation.

cat gencode.v45.primary\_assembly.annotation.gtf ERCC.gtf M81\_DFLR.chrEBV.inverted.gtf > gencode.v45.primary\_assembly.ERCC.M81\_DFLR.chrEBV.inverted.gtf

# Concatenate human, ERCC, and EBV fasta.

cat GRCh38.p14.genome.fa ERCC92.fa M81\_DFLR.chrEBV.inverted.fa > GRCh38.p14.ERCC.M81\_DFLR.chrEBV.inverted.fa

**STAR**

**Build STAR index**

# STAR version: 2.7.6a

# PMID: 23104886

STAR \

--runThreadN 20 \

--runMode genomeGenerate \

--genomeDir STAR\_index \

--genomeFastaFiles GRCh38.p14.ERCC.M81\_DFLR.chrEBV.inverted.fa \

--sjdbGTFfile gencode.v45.primary\_assembly.ERCC.M81\_DFLR.chrEBV.inverted.gtf \

--sjdbOverhang 100 \

--outFileNamePrefix STAR\_index

**Run STAR alignment**

STAR \

--twopassMode Basic \ # detects novel splice junctions

--genomeDir STAR\_index \

--outSAMunmapped Within \ # output unmapped reds within SAM/BAM

--outFilterType BySJout \ # filter reads with spurious junctions

--outSAMattributes NH HI AS NM MD MC \

--sjdbScore 1 \ # RSEM sets this to 1.

--readFilesCommand zcat \

--runThreadN 20 \

--outSAMtype BAM SortedByCoordinate \

--quantMode TranscriptomeSAM \ # transcriptome reads for RSEM

--outSAMheaderHD @HD VN:1.4 SO:coordinate \ # manually change header

--outFileNamePrefix sample\_id \

--outTmpeDir sample\_id\_tmp \

--readFilesIn forward.fastq reverse.fastq

**BASH script**

/data3/Alejandro/LCL\_project/scripts/RNAseq\_pipeline

**RSEM**

**Build RSEM index**

# RSEM version: 1.3.1

# PMID: 21816040

rsem-prepare-reference \

--gtf gencode.v45.primary\_assembly.ERCC.M81\_DFLR.chrEBV.inverted.gtf \

GRCh38.p14.ERCC.M81\_DFLR.chrEBV.inverted.fa \

rsem\_index

**Run RSEM transcript quantification**

rsem-calculate-expression \

--num-threads 20 \

--alignments \

--bam \

--paired-end \

--seed 12345 \

--estimate-rspd \

--no-bam-output \

--strandedness reverse \

sample\_id.Aligned.toTranscriptome.out.bam \

rsem\_index \

sample\_id

**BASH script**

~/lyticEBV\_RNAseq/Bash\_scripts/RNAseq\_pipeline.sh

**RNA-seq metrics**

**Software and versions**

# Java version: 21.0.1

# Picard CollectRnaSeqMetrics version: 3.1.0

# http://broadinstitute.github.io/picard/

# UCSC gtfToGenePred

# <https://genome.ucsc.edu/>

**Prepare filtered host references**

# Filter FASTA

samtools faidx GRCh38.p14.genome.fa “chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8 chr9 chr10 chr11 chr12 chr13 chr14 chr15 chr16 chr17 chr18 chr19 chr20 chr21 chr22 chrX chrY chrM” > host.fa

# Filter GTF

grep -E -w “chr1|chr2|chr3|chr4|chr5|chr6|chr7|chr8|chr9|chr10|chr11|chr12|chr13|chr14|chr15|chr16|chr17|chr18|chr19|chr20|chr21|chr22|chrX|chrY|chrM” > host.gtf

**Generate RefFlat using USCS gtfToGenePred**

# Generate host RefFlat file

gtfToGenePred -genePredExt -geneNameAsName2 -ignoreGroupsWithoutExons host.gtf /dev/stdout | awk ‘BEGIN { OFS=”\t” {print $12, $1, $2, $3, $4, $5, $6, $7, $8, $9, $10}’ > host.refflat

**Generate ribosomal intervals**

# Calculate chrom sizes

samtools faidx host.fa # index FASTA

cut -f1,2 host.fa.fai > chrom\_sizes

# rRNA interval\_list file suitable for Picard CollectRnaSeqMetrics

perl -lane ‘print “\@SQ\tSN:$F[0]\tLN:$F[1]\tAS:hg38”’ chrom\_sizes | grep -v \_ >> host.rRNA.interval\_list

grep ‘gene\_type “rRNA”’ host.gtf | awk ‘$3 == “transcript”’ | cut -f1,4,5,7,9 | perl -lane ‘/transcript\_id “([^”]+)”/ or die “no transcript\_id on $.”; print join “\t”, (@F[0,1,2,3], $1)’ | sort -k1V -k2n -k3n >> host.rRNA.interval\_list

**Prepare filtered sam headers**

# Prepare host bed files containing chromosome length info

cat host.fai | awk ‘BEGIN {OFS=”\t”} {print $1, 0, $2}’ > host.genome.bed

# Prepare the filtered host header

cat host.bed | awk ‘{print $1}’ | awk ‘$1=”SN:”$1’ > host.header\_grep.txt

# Filter the sam header (host) and reheader

samtools index sample.Aligned.sortedByCoord.out.bam

samtools view -H sample.Aligned.sortedByCoord.out.bam | grep -f host.header\_grep.txt > host.header.sam

samtools view -b -h -L host.bed sample.Aligned.sortedbyCoord.out.bam > sample.host.filtered.tmp.bam

samtools reheader host.header.sam sample.host.filtered.tmp.bam > sample.host.filtered.reheader.tmp.bam

**Run Picard CollectRnaSeqMetrics**

# host metrics

java -jar picard.jar CollectRnaSeqMetrics \

I=sample.host.filtered.reheader.tmp.bam \

O=sample.host.metrics \

REF\_FLAT= host.refflat \

STRAND=SECOND\_READ\_TRANSCRIPTION\_STRAND \

RIBOSOMAL\_INTERVALS= host.rRNA.interval\_list

**BASH script**

~/lyticEBV\_RNAseq/Bash\_scripts/collectmetrics.sh

**SparK**

**Install SparK**

# <https://github.com/harbourlab/SparK>

# Clone

git clone <https://github.com/harbourlab/SparK.git>

**Run**

# Run bamCoverage to get bedgraph files covering specific region.

# Using TBC1D10A readthrough transcription region as example

bamCoverage \

-b sample.Aligned.sortedByCoord.out.bam \

-of bedgraph

--numberOfProcessors 15 \

--region chr22:30262500-30304967 \

--scaleFactor <value from sampleinfo\_IGV.txt> # see DESeq2\_ERCCnorm\_SizeFactors.R script

--filterRNAstrand reverse \

-o TBC1D10A-R.bdg

bamCoverage \

-b sample.Aligned.sortedByCoord.out.bam \

-of bedgraph

--numberOfProcessors 15 \

--region chr22:30262500-30304967 \

--scaleFactor <value from sampleinfo\_IGV.txt> # see DESeq2\_ERCCnorm\_SizeFactors.R script

--filterRNAstrand forward \

-o sample.TBC1D10A-F.bdg

# Run bedtools to average bedGraph files by replicate

bedtools unionbedg -i WT\_03.TBC1D10A-F.bdg WT\_08.TBC1D10A-F.bdg WT\_13.TBC1D10A-F.bdg WT\_18.TBC1D10A-F.bdg WT\_23.TBC1D10A-F.bdg | awk 'OFS="\t" {sum=0; for (col=4; col<=NF; col++) sum += $col; print $1, $2, $3, sum/(NF-4+1); }' > WT\_Latent.avg.TBC1D10A-F.bdg

bedtools unionbedg -i WT\_03.TBC1D10A-R.bdg WT\_08.TBC1D10A-R.bdg WT\_13.TBC1D10A-R.bdg WT\_18.TBC1D10A-R.bdg WT\_23.TBC1D10A-R.bdg | awk 'OFS="\t" {sum=0; for (col=4; col<=NF; col++) sum += $col; print $1, $2, $3, sum/(NF-4+1); }' > WT\_Latent.avg.TBC1D10A-R.bdg

# Run SparK to generate forward and reverse tracks

python SparK.py \

-pr chr22:30262500-30304967 \

-cg 1 2 3 \

-gl Latent Early Late \

-gtf gencode.45.primary\_assembly.M81\_DFLR.chrEBV.inverted.ERCC.gtf \

-cf WT\_Latent.avg.TBC1D10A-F.bdg WT\_Early.avg.TBC1D10A-F.bdg WT\_Late.avg.TBC1D10A-F.bdg \

-o TBC1D10A\_RNAseq\_pileup\_F

python SparK.py \

-pr chr22:30262500-30304967 \

-cg 1 2 3 \

-gl Latent Early Late \

-gtf gencode.45.primary\_assembly.M81\_DFLR.chrEBV.inverted.ERCC.gtf \

-cf WT\_Latent.avg.TBC1D10A-R.bdg WT\_Early.avg.TBC1D10A-R.bdg WT\_Late.avg.TBC1D10A-R.bdg \

-o TBC1D10A\_RNAseq\_pileup\_R

**BASH script**

~/lyticEBV\_RNAseq/Bash\_scripts/bam2SparK\_v3.0.sh

**ARTDeco**

**Install ARTDeco**

# https://github.com/sjroth/ARTDeco

# Clone

git clone <https://github.com/sjroth/ARTDeco.git>

# Create environment

mamba env create -f ~/ARTDeco/Conda/environment.yml

# Installation

python ~/ARTDeco/setup.py install

**Prepare files**

# Concatenate

cat gencode.v45.primary\_assembly.annotation.gtf ERCC92.gtf | sed -e “s/\r//g” > gencode.v45.primary\_assembly.ERCC.gtf

cat GRCh38.p14.genome.fa ERCC92.fa | sed -e “s/\r//g” > GRCh38.p14.ERCC.fa

# Test GTF using BEDOPS gtf2bed

gtf2bed < gencode.v45.primary\_assembly.ERCC.gtf

# If the output is the following, fix with the code below: Error: Potentially missing gene or transcript ID from GTF attributes (malformed GTF at line [1]?)

awk '{ if ($0 ~ "transcript\_id") print $0; else print $0" transcript\_id \"\";"; }' gencode.v45.primary\_assembly.ERCC.gtf > gencode.v45.primary\_assembly.ERCC.modified\_genes.gtf

## Prepare chromosome sizes

samtools faidx GRCh38.p14.ERCC.fa

cut -f1,2 GRCh38.p14.ERCC.fa.fai > GRCh38.p14.ERCC.chrom.sizes

## Prepare location of BAM files

# Only genome BAM files must be present in directory

# Temporarily move genome alignment files to a separate directory

mv ~/STAR/\*Aligned.sortedByCoord.out.bam ~/STAR/genome

**Run ARTDeco**

# Run ARTDeco without differential expression analysis

ARTDeco \

-home-dir /ARTDeco/output \

-bam-files-dir /STAR/genome \

-gtf gencode.v45.primary\_assembly.ERCC.modified\_genes.gtf \

-cpu 20 \

-chrom-sizes-file GRCh38.p14.ERCC.chrom.sizes \

-layout PE \

-stranded True \

-orientation Reverse \

-skip-bam-summary

# Move genome alignment files back to original directory

mv ~/STAR/genome/\* ~/STAR

**BASH script**

~/lyticEBV\_RNAseq/Bash\_scripts/ARTDeco.sh

**IsoformSwitchAnalyzeR External Sequence Analysis**

**IsoformSwitchAnalyzeR**

# Run the IsoformSwitchAnalyzeR\_external.R script to get sequences required for external software

**CPC2 standalone**

# Download

wget https://github.com/gao-lab/CPC2\_standalone/archive/refs/tags/v1.0.1.tar.gz

# Unpack the tarbell

gzip -dc v1.0.1.tar.gz | tar xf –

# Build third-part packages

cd CPC2\_standalone-1.0.1

export CPC\_HOME="$PWD"

cd libs/libsvm

gzip -dc libsvm-3.18.tar.gz | tar xf -

libsvm-3.18

make clean && make

# Run locally using python3

~/CPC2\_standalone-1.0.1/bin/CPC2.py -i isoformSwitchAnalyzeR\_isoform\_nt.fasta -o cpc2output

**Pfam**

## Ran through Galaxy

# Search PfamScan and download Pfam-A.hmm.gz, Pfam-A.hmm.dat.gz, and active\_sites.dat.gz from the Required files.

# Unzip these files and upload to galaxy along with the isoformSwitchAnalyzeR\_isoform\_AA\_complete.fasta file

# Run

**IUPred2A**

# Ran on webserver: <https://iupred2a.elte.hu/>

# Submitted multiple subsets of the isoformSwitchAnalyzeR\_isoform\_AA fasta file by using the extractSequences( alsoSplitFastaFile=TRUE )

**SignalP-6.0**

# Ran locally

# Download software

# Go to <https://services.healthtech.dtu.dk/service.php?SignalP-6.0>

# Go to Downloads > Version 6.0h > fast > fill out information

# Use wget to download emailed hyperlink for signalp-6.0h.fast.tar.gz

# Unpack

tar xvzf signalp-6.0h.fast.tar.gz

# Create mamba environment and activate

mamba create -n py3.7 python=3.7

# Install

pip install signalp-6-package/

# Run

signalp6 --model\_dir ~/signalp6\_fast/signalp-6-package/models/ --fastafile isoformSwitchAnalyzeR\_isoform\_AA\_complete.fasta --organism eukarya --output\_dir /path/to/be/saved

**DeepLoc 2.0**

# Ran locally

# Download software

# Go to <https://services.healthtech.dtu.dk/services/DeepLoc-2.0/>

# Go to Downloads > Version 2.0 > all > fill out information

# Use wget to download emailed hyperlink for deeploc-2.0.All.tar.gz

# Create mamba environment and activate

mamba create -n py3.7 python=3.7

# Install

pip3 install deeploc-2.0.All.tar.gz

# Run

deeploc2 -f isoformSwitchAnalyzeR\_isoform\_AA\_complete.fasta -o deeploc\_results -m Accurate

**DeepTMHMM**

# Ran locally

# Install biolib

pip3 install pybiolib

# Note: the program kept timing out when running default via cloud.

# Note: when running locally, was receiving permission issues with Docker. This was fixed by following the recommendations from stackoverflow: <https://stackoverflow.com/questions/48957195/how-to-fix-docker-got-permission-denied-issue>

# Run:

biolib run --local DTU/DeepTMHMM --fasta isoformSwitchAnalyzeR\_isoform\_AA\_complete.fasta