ENCODE Oxford Nanopore LR-Split-seq Analysis Protocol
(v 1.)

Prepared by Fairlie Reese and Jasmine Sakr
July 27, 2022
Mortazavi Lab, University of California, Irvine

Contact Information

Fairlie Reese
2300 Biological Sciences III
University of California Irvine
Irvine, CA 92697-2300
Telephone: (949) 824-8393
Email: freese@uci.edu

Jasmine Sakr
2300 Biological Sciences III
University of California, Irvine
Irvine, CA 92697-2300
Telephone: (949) 824-8393
Email: jsakr@uci.edu

Ali Mortazavi
2218 Biological Sciences III
University of California Irvine
Irvine, CA 92697-2300
Telephone: (949) 824-6762
Email: ali.mortazavi@uci.edu

I. Overview

The LR-Split-seq data on the ENCODE portal was processed in the lab and submitted to the portal. This document describes the steps taken to process the data in detail. The software versions used can be found in Table 1.

Processing is done using Oxford Nanopore's (ONT) software. The ONT sequencer writes out the raw signal data in a series of fast5 files that each contain 4,000 reads. The signal is converted into a fastq of sequences using ONT's guppy basecaller.

Reads output from Guppy are then demultiplexed for their Split-seq barcodes using custom code termed LR-splitpipe, which is adapted from the Parse Biosciences data processing code. **The fastq submitted to the portal is an intermediate output file from LR-splitpipe which consists of all the reads that contain a valid first 8nt combinatorial barcode which identifies the sample.** In the rest of the LR-splitpipe pipeline, the second and third combinatorial barcodes are identified and concatenated into a 24nt cell barcode which is then added to the read name for each read and output in a fastq. Reads are then mapped to the genome using minimap2 LR-splitpipe. Reads are then mapped using Minimap2 to yield read alignments in sam format, and corrected for long-read sequencing artifacts such as microindels and noncanonical splice junctions using TranscriptClean. At this point, an additional LR-splitpipe module is run to remove the 24nt cell barcodes from each read name and add it to the sam / bam file as a CB sam tag. **This bam file is the one that's on the portal, again demultiplexed for each sample.**

Name	Version	Available from
Guppy	6.0.1	N/A
samtools	1.10	https://sourceforge.net/projects/samtools/files/samtools/1.10/
LR-splitpipe	v2.0	https://github.com/fairliereese/LR-splitpipe
Minimap2	2.17-r94	https://github.com/lh3/minimap2
TranscriptClean	2.0.2	https://github.com/dewyman/TranscriptClean

II. Computational analysis

A. Basecalling reads with Guppy

ONT outputs the signal for each read in **fast5** format. This raw signal must be converted into **fastq** format using basecalling.

```
ccs \
   --skip-polish \
   --min-length 10 \
   --min-passes 3 \
   --min-rq=0.9 \
```

```
--min-snr=2.5 \
--report-file ccs_report.txt \
ccs.bam
```

Each CCS run produces a bam file, ccs.bam. The sequences in these files are reads of insert (ROIs), which represent the consensus sequence of each read.

C. Demultiplexing reads with LR-splitpipe

Reads are then demultiplexed to identify which cell each read comes from. This is done with the LR-splitpipe (available here: https://github.com/fairliereese/LR-splitpipe) custom code which is modeled off of the Parse Biosciences short-read demultiplexing strategy. LR-splitpipe was run using the following parameters:

```
python demultiplex.py all \
        -f flnc.fastq \
        -o <output file prefix> \
        -t 64 \
        –l1_mm 3 \
        –l2_mm 3 \
        –chunksize 2000000 \
        —verbosity 2 \
        —delete_input
```

After generating the fastq with cell barcodes for each read, the fastq was split into individual fastqs representative of each sample based on the identity of the sample associated with the barcode. **This split fastq is the one that is on the portal.** This approach allows for portal users to possibly improve on the demultiplexing in the future while still separating the data out by input sample.

D. Extract reference splice junctions

TranscriptClean / Minimap2 require a file of reference splice junctions in order to correct noncanonical junctions in the PacBio transcripts. To get this file, we run a TranscriptClean utility on the GENCODE vM21 (for mouse) or GENCODE v29 (for human) comprehensive gene annotation (reference chromosomes only).
These files are available here:
https://www.gencodegenes.org/human/release_29.html
https://www.gencodegenes.org/mouse/release_M21.html

Using the human references:

```
python ${transcriptclean_path}/accessory_scripts/get_SJs_from_gtf.py \
   --f ../gencode.v29.annotation.gtf \
   --g GRCh38.fa \
   --o gencode_v29_SJs.tsv
```

Using the mouse references:

```
python ${transcriptclean_path}/accessory_scripts/get_SJs_from_gtf.py \
   --f ../gencode.vM21.annotation.gtf \
   --g mm10.fa \
   --o gencode_vM21_SJs.tsv
```

The output file **gencode_v##_SJs.tsv**, contains splice junctions derived from the reference transcriptome. For each splice junction, it lists genomic location, strand, intron motif, and two additional placeholder columns (to match formatting to a type of STAR splice junction file).

E. Alignment to the reference genome

We next align the FLNC reads to the GRCh38 XY (for human) and mm10 (for mouse) reference genome (https://www.encodeproject.org/data-standards/reference-sequences/). Run Minimap2 with the following parameters:

For human samples:

```
minimap2 \
        -t 16 \
        -ax splice:hq \
        -uf \
        --MD \
       GRCh38.fa \
       flnc_dmux.fastq \
       > aligned.out.sam
```

For mouse samples:

```
minimap2 -t 16 \
        -ax splice:hq \
        -uf \
        --MD \
       mm10.fa \
       flnc_dmux.fastq \
       > aligned.out.sam
```

The output file will be called **aligned.out.sam**.

F. Reference-based error correction with TranscriptClean

Although the CCS process catches many of the errors found in PacBio transcripts, longer reads and/or those with fewer passes are still prone to mismatch and microindel errors. If these occur on the boundary of an intron, they may create the mistaken appearance of a novel splice junction. TranscriptClean is a Python program we developed to compare the sequences of mapped isoforms to the reference genome and correct likely errors. It can be downloaded from Github at https://github.com/dewyman/TranscriptClean. Run TranscriptClean version 2.0.2 on the FLNC reads using the parameters below.

For human datasets:

```
python ${transcriptclean_path}TranscriptClean.py \
   --sam aligned.out.sam \
   --genome GRCh38.fa \
   --spliceJns gencode_v29_SJs.tsv \
   --primaryOnly \
   --canonOnly \
   --deleteTmp \
   --outprefix libraryID
```

For mouse datasets:

```
python ${transcriptclean_path}TranscriptClean.py \
   --sam aligned.out.sam \
   --genome mm10.fa \
   --spliceJns gencode_vM21_SJs.tsv \
   --primaryOnly \
   --canonOnly \
   --deleteTmp \
   --outprefix libraryID
```

This command will generate two output files: **libraryID_clean.sam** and **libraryID_clean.fa**. These contain the reads with corrections made to remove microindels, mismatches, and noncanonical splice junctions as specified by the parameters. The inclusion of the --canonOnly parameter ensures that all of the reads in the output contain only canonical splice junctions or noncanonical splice junctions that are supported by the reference annotation.

G. Reformat reads to add cell barcode tags

After running Minimap2 and TranscriptClean, reads are now in aligned sam format and the cell barcodes, which were appended to the read ID of each read in the first LR-splitpipe step, can now be added as **CB** sam tags, which are traditionally reserved for cell barcodes. Additionally, this step merges the random hexamer and oligo-dT primed barcode 1s that belong to the same well. This is done using an additional script included in the LR-splitpipe repository as follows:

```
python add_bam_tag.py \
```

**-s clean.sam **
**--merge_primers **
**--suffix <suffix> **
-o <output file prefix>

The output file will be called **<output file prefix>_merged_primers.sam.**

III. References

1. Li, H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics 34, 3094–3100 (2018).
2. Gordon, S. P. et al. Widespread Polycistronic Transcripts in Fungi Revealed by Single-Molecule mRNA Sequencing. PLoS ONE 10, e0132628 (2015).
3. Li H.*, Handsaker B.*, Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R. and 1000 Genome Project Data Processing Subgroup (2009) The Sequence alignment/map (SAM) format and SAMtools. Bioinformatics, 25, 2078-9.
4. Wyman, D., TranscriptClean: A program for correcting mismatches, microindels, and noncanonical splice junctions in long reads, (2018), GitHub repository, https://github.com/dewyman/TranscriptClean
5. Reese, F. LR-splitpipe (2021), GitHub repository https://github.com/fairliereese/LR-splitpipe