Starling-May18

Projects/Katarina Stuart/KStuart.Starling-Aug18/Sv3_Genome/Transcriptome/2020-04-04.TamaIsoseq

PDF Version generated by

Katarina Stuart (z5188231@ad.unsw.edu.au)

on

Jun 23, 2022 @03:47 PM NZST

Table of Contents

2020-04-04.TamaIsoseq

Katarina Stuart (z5188231@ad.unsw.edu.au) - Jun 23, 2022, 3:47 PM NZST

Aligning to the final starling assembly, Merging Isoforms

Minimap2:

https://github.com/Magdoll/cDNA_Cupcake/wiki/Best-practice-for-aligning-Iso-Seq-to-reference-genome:-minimap2,-deSALT,-GMAP,-STAR,-BLAT

Using minimap2 to align reads to a genome

NOTE: please use version 2.9 and above so it will support the `--secondary=no` option.

A usage example would be:

```
minimap2 -t 30 -ax splice -uf --secondary=no -C5 -O6,24 -B4 \
   hg38.fasta hq_isoforms.fasta \
   > hq_isoforms.fasta.sam \
  2> hq_isoforms.fasta.sam.log
```

which would use 30 CPUs, spliced alignment, SAM output, and trust the orientation of the provided sequence (since Iso-Seq data is already orientated using primers and polyA tails). The `--secondary=no` option means only the best alignment will be output, which is required for running the Cupcake `collapse_isoforms_by_sam.py` script later.

For organisms that may use non-canonical GT/AG splice junctions, consider using `-C5` or `--splice-flank=no -C5`. See this minimap2 GitHub issue:99 for more explanation.

We recently tested adding parameters `-O6,24 -B4` and found that it could align more known exons in human test data, hence we recommend using that as well.

minimap2 supports both an un-indexed reference fasta (ex: `hg38.fasta`) or you can prebuilt an index to speed up alignment:

```
minimap2 -d hg38.mmi hg38.fasta
```

Note that the strand is encoded using the FLAG field (see minimap2 issue:88) which is already properly handled by the Cupcake collapse_isoforms_by_sam.py script.

Script

```
#!/bin/bash

#PBS -N 2020-04-14.IsoseqMinimap.pbs
#PBS -l nodes=1:ppn=16
#PBS -l mem=124gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -M katarina.stuart@student.unsw.edu.au
#PBS -m ae

module purge
module load minimap2/2.17
```

```
cd /srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.1_StarlingIsoseq/mapping/minimap_3.2.1

GENOME=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/genome_assembly/Sturnus_vulgaris_2.3.1.simp.fasta
ISOSEQ=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.1_StarlingIsoseq/analysis/Isoseq3.3_pipeline/polya_8/clustered.hq.fasta

minimap2 -t 16 -ax splice -uf --secondary=no --splice-flank=no -C5 -O6,24 -B4 \
    ${GENOME} ${ISOSEQ} \
    > clustered.hq.fasta.sam \
    2> clustered.hq.fasta.sam.log
```

Getting Bam:

Tama Collapse

Collapsing redundant isoform information

https://github.com/GenomeRIK/tama/wiki/Tama-Collapse

Manual

usage: tama_collapse.py [-h] [-s] [-f] [-p] [-x] [-e] [-c] [-i] [-a] [-m] [-z]

This script collapses mapped transcript models

arguments:

```
  -h, --help  show this help message and exit
  -s S        Sorted sam/bam file (required)(if using BAM file please use -b BAM flag as well)
  -f F        Fasta file (required)
  -p P        Output prefix (required)
  -x X        Capped flag: capped or no_cap
  -e E        Collapse exon ends flag: common_ends or longest_ends (default
              common_ends)
  -c C        Coverage (default 99)
  -i I        Identity (default 85)
  -icm ICM    Identity calculation method (default ident_cov for including coverage) (alternate is ident_map
for excluding hard and soft clipping)
  -a A        5 prime threshold (default 10)
  -m M        Exon/Splice junction threshold (default 10)
  -z Z        3 prime threshold (default 10)
  -d D        Flag for merging duplicate transcript groups (default is merge_dup will merge duplicates
,no_merge quits when duplicates are found)
  -sj SJ      Use error threshold to prioritize the use of splice junction information from collapsing
transcripts(default no_priority, activate with sj_priority)
  -sjt SJT    Threshold for detecting errors near splice junctions (default is 10bp)
  -lde LDE    Threshold for amount of local density error near splice junctions that is allowed (default is
1000 errors which practically means no threshold is applied)
  -ses SES    Simple error symbol. Use this to pick the symbol used to represent matches in the simple error
string for LDE output.
  -b   BAM    Use BAM instead of SAM
  -log LOG    Turns on/off output of collapsing process. (default on, use log_off to turn off)
```

Default command would look like this:

python tama_collapse.py -s mapped_reads.sam -f genome.fa -p prefix -x capped

script

```
module add python/2.7.15
module load bowtie/2.3.5.1

cd /srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.1_StarlingIsoseq/mapping/minimap_3.2.1

TAMA=/srv/scratch/z5188231/KStuart.Starling-Aug18/programs/tama-master
GENOME=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/genome_assembly/Sturnus_vulgaris_2.3.1.simp.fasta
ISOSEQ=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.1_StarlingIsoseq/analysis/Isoseq3.3_pipeline/polya_8/clustered.hq.fasta
```

```
samtools sort clustered.hq.fasta.sam -o clustered.hq.fasta_sort.sam
```

```
python ${TAMA}/tama_collapse.py -s clustered.hq.fasta_sort.sam -f ${GENOME} -p Svulagris -x capped
```

Version two with tweaks as per yuanyuan's recommendation

With tama_collapse, I suggest running it with the sj_priority option, and it might be useful to test a few different settings for 5' (-a) and 3' (-z) collapsing and see how much difference each makes. I usually use something like this:

python ${TAMA}/tama_collapse.py -f ${GENOME} -x capped -p ${prefix} -a 100 -z 30 -sj sj_priority -lde 5

- sj sj_priority is used to rank evidence for splice junction prediction based on each read's sequence identity to the reference surrounding the splice junction;
- lde 5 filters out reads that have more than 5 mismatches within 10bp on either side of a splice junction (there should not be too many reads getting filtered out by this filter, as you are using polished transcripts as input for mapping).
These help improve accuracy of splice junction annotation.

You might want to do a little sanity check in IGV to make sure the mapping and predicted gene models look reasonable (e.g. sometimes you might need to reduce the maximum allowed intron size with minimap2). Once you are happy with the annotation, you can then use the getfasta function from bedtools to extract the final set of unique transcript sequences:

bedtools getfasta -fi ${GENOME} -bed tama_collapse_output.bed -fo output.fasta -s -name -split

```
module add python/2.7.15
module load bowtie/2.3.5.1

cd /srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.1_StarlingIsoseq/mapping/minimap_3.2.1

TAMA=/srv/scratch/z5188231/KStuart.Starling-Aug18/programs/tama-master
GENOME=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/genome_assembly/Sturnus_vulgaris_2.3.1.simp.fasta
PREFIX=Starling
```

```
python ${TAMA}/tama_collapse.py -s clustered.hq.fasta_sort.sam -f ${GENOME} -x capped -p ${PREFIX}.a100.z30 -a 100 -z 30 -sj sj_priority -lde 5 #yuanyuan recommendation
```

```
python ${TAMA}/tama_collapse.py -s clustered.hq.fasta_sort.sam -f ${GENOME} -x capped -p ${PREFIX}.a50.z20 -a 50 -z 20 -sj sj_priority -lde 5 #less stringent
```

```
python ${TAMA}/tama_collapse.py -s clustered.hq.fasta_sort.sam -f ${GENOME} -x capped -p ${PREFIX}.a150.z40 -a 150 -z 40 -sj sj_priority -lde 5 #more stringent
```

grep -o -i discarded Svulagris_read.txt | wc -l

2653

grep -o -i discarded Starling.a50.z20_read.txt | wc -l
2653

grep -o -i discarded Starling.a100.z30_read.txt | wc -l
2653

grep -o -i discarded Starling.a150.z40_read.txt | wc -l
2653

less Svulagris_trans_report.txt | wc -l
29532

less Starling.a50.z20_trans_report.txt | wc -l
28927

less Starling.a100.z30_trans_report.txt | wc -l
28178

less Starling.a150.z40_trans_report.txt | wc -l
27071

```
module add bedtools
bedtools getfasta -fi ${GENOME} -bed Starling.a100.z30.bed -fo Starling.a100.z30.fasta -s -name -split
bedtools getfasta -fi ${GENOME} -bed Starling.a150.z40.bed -fo Starling.a150.z40.fasta -s -name -split

bedtools getfasta -fi ${GENOME} -bed Starling.a50.z20.bed -fo Starling.a50.z20.fasta -s -name -split

bedtools getfasta -fi ${GENOME} -bed Svulagris.bed -fo Svulagris.fasta -s -name -split
```

grep "^>" Svulagris.a100.z30.fasta | wc -l

grep "^>" Svulagris.a150.z40.fasta | wc -l

grep "^>" Svulagris.a50.z20.fasta | wc -l

grep "^>" Svulagris.fasta | wc -l

you could use IGV to manually examine some genes which you know the structure of, to make sure things make sense and all the tools have worked properly. One thing to watch out for is prevalence of large introns due to mapping artefacts – if this happens you can usually spot it easily by just skimming through gene models for a number of scaffolds (of course you can do this more thoroughly by checking the bed files directly). If lots of predicted genes have huge introns, you might need to adjust mapping parameters.

FINAL TRANSCRIPTOME FILE: Svulagris.a100.z30.fasta

Trying with stricter intron length to see if it affects gene prediction models

Usage: minimap2 [options] <target.fa>|<target.idx> [query.fa] [...]
Options:
  Indexing:

```
  -H          use homopolymer-compressed k-mer (preferrable for PacBio)
  -k INT      k-mer size (no larger than 28) [15]
  -w INT      minimizer window size [10]
  -I NUM      split index for every ~NUM input bases [4G]
  -d FILE     dump index to FILE []
Mapping:
  -f FLOAT    filter out top FLOAT fraction of repetitive minimizers [0.0002]
  -g NUM      stop chain enlongation if there are no minimizers in INT-bp [5000]
  -G NUM      max intron length (effective with -xsplice; changing -r) [200k]
  -F NUM      max fragment length (effective with -xsr or in the fragment mode) [800]
  -r NUM      bandwidth used in chaining and DP-based alignment [500]
  -n INT      minimal number of minimizers on a chain [3]
  -m INT      minimal chaining score (matching bases minus log gap penalty) [40]
  -X          skip self and dual mappings (for the all-vs-all mode)
  -p FLOAT    min secondary-to-primary score ratio [0.8]
  -N INT      retain at most INT secondary alignments [5]
Alignment:
  -A INT      matching score [2]
  -B INT      mismatch penalty [4]
  -O INT[,INT] gap open penalty [4,24]
  -E INT[,INT] gap extension penalty; a k-long gap costs min{O1+k*E1,O2+k*E2} [2,1]
  -z INT[,INT] Z-drop score and inversion Z-drop score [400,200]
  -s INT      minimal peak DP alignment score [80]
  -u CHAR     how to find GT-AG. f:transcript strand, b:both strands, n:don't match GT-AG [n]
Input/Output:
  -a          output in the SAM format (PAF by default)
  -o FILE     output alignments to FILE [stdout]
  -L          write CIGAR with >65535 ops at the CG tag
  -R STR      SAM read group line in a format like '@RG\tID:foo\tSM:bar' []
  -c          output CIGAR in PAF
  --cs[=STR]  output the cs tag; STR is 'short' (if absent) or 'long' [none]
  --MD        output the MD tag
  --eqx       write =/X CIGAR operators
  -Y          use soft clipping for supplementary alignments
  -t INT      number of threads [3]
  -K NUM      minibatch size for mapping [500M]
  --version   show version number
Preset:
  -x STR      preset (always applied before other options; see minimap2.1 for details) []
              - map-pb/map-ont: PacBio/Nanopore vs reference mapping
              - ava-pb/ava-ont: PacBio/Nanopore read overlap
              - asm5/asm10/asm20: asm-to-ref mapping, for ~0.1/1/5% sequence divergence
              - splice: long-read spliced alignment
              - sr: genomic short-read mapping
```

```
cd /srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.1_StarlingIsoseq/mapping/minimap_strict_introns
```

```
module purge
module load minimap2/2.17

GENOME=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/genome_assembly/Sturnus_vulgaris_2.2.simp.fasta
ISOSEQ=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.1_StarlingIsoseq/analysis/Isoseq3.3_pipeline/polya_8/clustered.hq.fasta

minimap2 -t 16 -ax splice -uf --secondary=no --splice-flank=no -C5 -O6,24 -B4 -G 100k\
    ${GENOME} ${ISOSEQ} \
    > clustered.hq.fasta.intron100.sam \
    2> clustered.hq.fasta.intron100.sam.log
```

```
module add python/2.7.15
module load bowtie/2.3.5.1

TAMA=/home/z5188231/programs/tama-master
GENOME=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/genome_assembly/Sturnus_vulgaris_2.2.simp.fasta
PREFIX=Starling

samtools sort clustered.hq.fasta.intron100.sam -o clustered.hq.fasta.intron100_sort.sam

python ${TAMA}/tama_collapse.py -s clustered.hq.fasta.intron100_sort.sam -f ${GENOME} -x capped -p ${PREFIX}.a100.z30 -a 100 -z 30 -sj sj_priority -lde 5 #yuanyuan recommendation



minimap2 -t 16 -ax splice -uf --secondary=no --splice-flank=no -C5 -O6,24 -B4 -G 50k\
    ${GENOME} ${ISOSEQ} \
    > clustered.hq.fasta.intron50.sam \
    2> clustered.hq.fasta.intron50.sam.log


samtools sort clustered.hq.fasta.intron50.sam -o clustered.hq.fasta.intron50_sort.sam

python ${TAMA}/tama_collapse.py -s clustered.hq.fasta.intron50_sort.sam -f ${GENOME} -x capped -p ${PREFIX}.a100.z30.intron50 -a 100 -z 30 -sj sj_priority -lde 5 #yuanyuan recommendation
```

Checked the results in IGV and the gene structures look the same from intron max lenghs of 50, 100 and 200 (defualt).