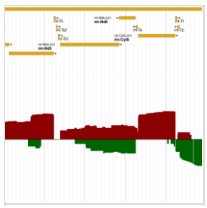


9

Mar 18, 2022

Stranded Mapping from Oriented Long Reads V.9

David A Eccles[1]

[1]Malaghan Institute of Medical Research (NZ)

1

dx.doi.org/10.17504/protocols.io.yxmvm45nv3pe/v9

High molecular weight DNA extraction from all kingdoms
Tech. support email: **See@each.protocol**


David Eccles
Malaghan Institute of Medical Research (NZ)

This protocol demonstrates how to map strand-oriented long reads to a genome, and visualise them in a genome browser.

The general idea is to use minimap2 to create stranded BAM files, which are split for forward/reverse orientation then converted into BigWig format for display in a genome browser.

Input(s):
- stranded fastq files (see protocol Preparing Reads for Stranded Mapping)
- a FASTA file containing the genome / sequence of interest.

Output(s):
- Genome-mapped stranded BAM files
- Genome-mapped stranded BigWig files

DOI

dx.doi.org/10.17504/protocols.io.yxmvm45nv3pe/v9

https://bioinformatics.stackexchange.com/a/3922/73

David A Eccles 2022. Stranded Mapping from Oriented Long Reads.
protocols.io
https://dx.doi.org/10.17504/protocols.io.yxmvm45nv3pe/v9
David Eccles

genome, mapping, transcript, nanopore, long reads

protocol ,

Mar 18, 2022

Mar 18, 2022

59598

You will need access to the following free and open-source software program(s):
- minimap2
- samtools

And the following additional data file(s):
- a FASTA file containing the genome / sequence of interest.

Orient Reads

1 Orient reads as per protocol Preparing Reads for Stranded Mapping.

If this has been done, then the following command should produce output without errors:

```
for bc in $(awk '{print $2}' barcode_counts.txt);
  do ls oriented/${bc}_reads_dirAdjusted.fq.gz;
done
```

Example output:

```
oriented/BC03_reads_dirAdjusted.fq.gz
oriented/BC04_reads_dirAdjusted.fq.gz
oriented/BC05_reads_dirAdjusted.fq.gz
oriented/BC06_reads_dirAdjusted.fq.gz
oriented/BC07_reads_dirAdjusted.fq.gz
oriented/BC08_reads_dirAdjusted.fq.gz
```

Index Preparation

2 Prepare genome index for spliced alignment:

minimap2 2.24
Linux
source by Heng Li

```
minimap2 -d GRCm39.primary_assembly.genome-splice.idx -Q -t 10 -x
splice GRCm39.primary_assembly.genome.fa
```


Citation: David A Eccles Stranded Mapping from Oriented Long Reads https://dx.doi.org/10.17504/protocols.io.yxmvm45nv3pe/v9

Prepare a chromosome size index file:

```
samtools faidx GRCm39.primary_assembly.genome.fa
cat GRCm39.primary_assembly.genome.fa.fai | awk '{print $1"\t"$2}'
> GRCm39.primary_assembly.genome.chrInfo.txt
```

Read Mapping

3 Map the long reads to the genome using minimap2, using samtools to covert to a sorted BAM format. This is where the reverse complementing done during demultiplexing gives a big saving of effort. As this BAM file is one of the main outputs, the run name is added to the file name.

SAMtools 1.8
Linux
source by Wellcome Trust Sanger Institute

```
runName="CHANGE_THIS";
indexFile="/index/location/GRCm39.primary_assembly.genome-
splice.idx";
mkdir -p mapped;
for bc in $(awk '{print $2}' barcode_counts.txt);
  do echo ${bc};
  minimap2 -t 10 -a -x splice ${indexFile}
oriented/${bc}_reads_dirAdjusted.fq.gz | \
    samtools view -b | samtools sort >
mapped/mm2_${runName}_called_${bc}_vs_genome.bam;
done
```

Creating BigWig Coverage Files (not needed for JBrowse 2)

4 **mpileupDC.pl**

A bedGraph of coverage is created using BEDTools. The default options for BEDTools treat sequence deletions (which happen frequently in nanopore reads) as a drop in coverage, which can make exon hunting and coverage calculation more difficult. I submitted a pull request to add an additional *ignoreD* parameter to the command line to allow cDNA reads with split coverage across introns to ignore deletions when considering coverage; this request has now been incorporated into the main BEDtools repository (as of v2.30.0):

Citation: David A Eccles Stranded Mapping from Oriented Long Reads https://dx.doi.org/10.17504/protocols.io.yxmvm45nv3pe/v9

BEDtools 2.30.0

Linux

source by Aaron Quinlan

```
runName="CHANGE_THIS";
for bc in $(awk '{print $2}' barcode_counts.txt);
  do echo ${bc};
  basename="mapped/mm2_${runName}_called_${bc}_vs_genome";
  samtools view -b ${basename}.bam | \
    bedtools genomecov -bga -strand '+' -split -ignoreD -ibam - >
${basename}.bg.plus
  samtools view -b ${basename}.bam | \
    bedtools genomecov -bga -strand '-' -split -ignoreD -ibam - >
${basename}.bg.minus
  perl -i -pe 's/([0-9]+)$/-$1/' ${basename}.bg.minus
done;
```

5 Stranded bedgraph files are converted to bigwig. This requires BEDTools and a genome information file containing chromosome lengths, as generated in the first step:

```
runName="CHANGE_THIS";
infoFile="/index/location/GRCm39.primary_assembly.genome-
splice.chrInfo.txt";
for bc in $(awk '{print $2}' barcode_counts.txt);
  do echo ${bc};
  basename="mapped/mm2_${runName}_called_${bc}_vs_genome"
  bedGraphToBigWig ${basename}.bg.plus ${infoFile}
${basename}.bw.plus
  bedGraphToBigWig ${basename}.bg.minus ${infoFile}
${basename}.bw.minus
done
```

JBrowse 2 Configuration

6

Citation: David A Eccles Stranded Mapping from Oriented Long Reads https://dx.doi.org/10.17504/protocols.io.yxmvm45nv3pe/v9

JBrowse 2 1.6.7

Linux

source by GMOD

Each BAM file should have its own JBrowse rack. JBrowse 2 can create both alignment views and coverage plots from BAM files:

```
runName="CHANGE_THIS";
jbrowseLoc="/path/to/jbrowse2/folder";
for bc in $(awk '{print $2}' barcode_counts.txt);
  do echo ${bc};
  basename="mapped/mm2_${runName}_called_${bc}_vs_genome";
  # create index
  samtools index ${basename}.bam;
  # copy and create track in Jbrowse2 config file
  jbrowse add-track ${basename}.bam --load copy --out ${jbrowseLoc}
--subDir bam
done;
```

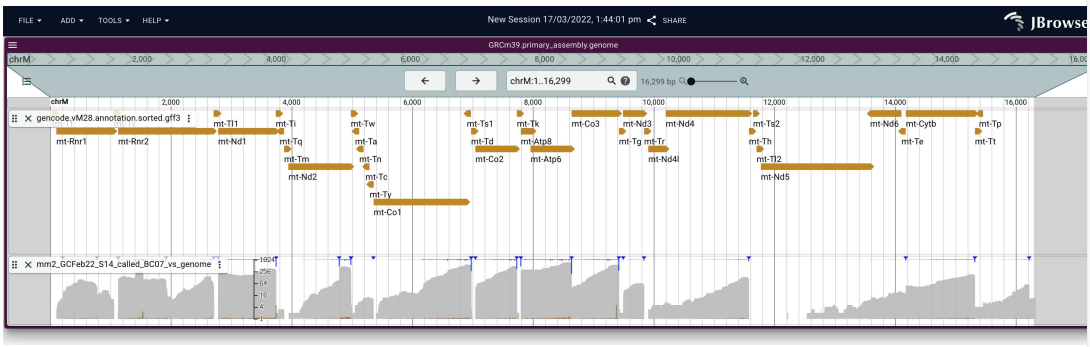
If you have a "smart" file system mount that doesn't let you directly copy files via the jbrowse add-track command, you can use the /tmp file system and symlinks to do the same thing:

```
runName="CHANGE_THIS";
jbrowseLoc="/path/to/jbrowse2/folder";
mkdir -p /tmp/jbrowse/bam
rm -v /tmp/jbrowse/bam/*
mkdir -p ${jbrowseLoc}/bam
for bc in $(awk '{print $2}' barcode_counts.txt);
  do echo ${bc};
  basename="mapped/mm2_${runName}_called_${bc}_vs_genome";
  # create index
  samtools index ${basename}.bam;
  # copy JBrowse2 config file to temporary folder
  cp -v ${jbrowseLoc}/config.json /tmp/jbrowse/
  # copy and create track in Jbrowse2 config file
  jbrowse add-track ${basename}.bam --load symlink --out
/tmp/jbrowse --subDir bam
  # copy BAM files and updated config file to JBrowse 2 folder
  cp -v /tmp/jbrowse/config.json ${jbrowseLoc}
  cp -v ${basename}.bam ${basename}.bam.bai ${jbrowseLoc}/bam
done;
```


Citation: David A Eccles Stranded Mapping from Oriented Long Reads https://dx.doi.org/10.17504/protocols.io.yxmvm45nv3pe/v9

Sanity Check

7 If this has worked properly, then mapping human or mouse to the mitochondrial genome with a LinearSNPCoverageDisplay in log scale should show a stepped expression, a bit like the *Expected Results* shown here:



Citation: David A Eccles Stranded Mapping from Oriented Long Reads https://dx.doi.org/10.17504/protocols.io.yxmvm45nv3pe/v9