Oct 04, 2024

RNAseq DEG analysis protocol for non-human samples using ONT reads

DOI
dx.doi.org/10.17504/protocols.io.5qpvokmbbl4o/v1

Avni Arora[1]

[1]NC State

Avni Arora
NC State

OPEN ACCESS

DOI: dx.doi.org/10.17504/protocols.io.5qpvokmbbl4o/v1

Protocol Citation: Avni Arora 2024. RNAseq DEG analysis protocol for non-human samples using ONT reads. **protocols.io** **https://dx.doi.org/10.17504/protocols.io.5qpvokmbbl4o/v1**

Protocol status: Working
We use this protocol and it's working

Created: October 02, 2024

Last Modified: October 04, 2024

Protocol Integer ID: 108964

Abstract

Data analysis of short-read sequencing from ONT flow cells can be done in a multitude of ways depending on the type of analysis that needs to be done and the sequencer utilized for assembly. Packages like NanoPack are available for use via Python however, there are fewer short-read sequencing packages available in RStudio. This protocol demonstrates how to analyze short-read transcriptome data in Rstudio and Python without the use of cloud-based software like EPI2ME.

Attachments

Protocol.pdf

45KB

Guidelines

This protocol was developed for the purposes of BIT 495: Special Topics in Biotechnology: Portable Genome Sequencing at North Carolina State University.

Materials

Rstudio
Python/access to condo environment

Before start

You may need to activate a conda environment and/or install python as well as RStudio in order to run these commands.

Data clean up and alignment

1 Obtain FASTQ files from flow cell (GridION, MinION, PromethION)

2 Upload FASTQ files to machine

Trimming

3 Utilize the Prowler options to build a script to trim your reads based on the adapter lengths produced by the Fasta file.

Note

ONT pipeline utilizes guppy software to trim the reads, this protocol recommends the use of Prowler, a novel trimming tool recently developed for Python.

Command

Prowler Trimming command

```
python3 TrimmerLarge.py -f [filename] -i [inFolder] -o [outFolder] -w [windowSize] -l [minLen] -c [clipping] -g [fragments] -m [mode] -q [Qcutoff] -d [maxDataMB] -r [outMode]
```

3.1

Command

Prowler command line options

```
-f,     --file,                filename:  The name of the file you
want to trim,
-i,     --infolder,     inFolder:The folderpath where your file to be
trimmed is located (default = cwd)
-o,     --outfolder,    outFolder: The folderpath where your want to
save the trimmed file (default = cwd)
-w,     --windowsize,   windowSize:  Change the size of the trimming
window (default= 100bp)
-l,     --minlen,                        minLen:  Change the minimum
acceptable numer of bases in a read (default=100)
-m,     --trimmode,     mode:           Select trimming algorithm: S
for static  or D for dynamic (default=S)
-q,     --qscore,       Qcutoff:        Select the phred quality
score trimming threshold (default=7)
-d,     --datamax,      maxDataMB:      Select a maximum data
subsample in MB (default=0, entire file)
-r,     --outformat,    outMode:        Select output format of
trimmed file (fastq or fasta) (default=.fastq)
-c,     --clip,         clipping:       Select L to clip leading Ns,
T to trim trialing Ns and LT to trim both (default=LT)
-g,     --fragments,    fragments:      Select fragmentation mode
(default=U0)


[filename] = filename, including file extention (string)
[inFolder] = source folder (string)
[outFolder] = output folder (string)
[windowSize] = number of bases in trimmer window (integer)
[minLen] = minimum number of bases in read. reads with fewer bases
will be rejected
[trimSpecs] = trim specs: [X1]-[X2]-[X3] (string)
   X1: "L", "T", "LT", or "". L clips leading Ns, T clips trailing
Ns, LT clips both, "" clips neither. recommended mode is LT
   X2: U0, F0, F1, F2... U0=unfragmented output, F0=fragmented output
with all fragments, F[n]= fragmented output with n largest
fragments
   X3: S, D. S=static mode. D=dynamic mode

   example: "X1-X2-X3" -> "LT-U0-S"
```

```
[Qcutoff] = phred quality score threshold. (integer)
[seqsToAnalyze] = megabytes of data to trim. trimmer will read files
in 1 MB chunks and stop when number of MB exceeds this number.
[outMode]: output file extension. saves trimmed data as either
".fasta" or ".fastq"
```

Alignment and Quantification

4

Note

Minimap2 is an alignment software developed to align short reads produced by ONT flow cells to a reference genome. Minimap2 has been proven to be faster and more accurate than other short read alignment softwares on the market currently. As the following quantification step via Salmon requires an input of SAM files, be sure to tell the program to output SAM files. Your input files should be gzipped .fasta, .fa, or .fq files obtained from the Prowler output.

5 Creating reference index

Note

In order to align the reads to a reference genome, you must create an index containing the reference genome of interest. Adjust demo script as needed to include/exclude options. Download the reference FASTA file for your organism of interest from the appropriate repository (eg. Flybase, Mouse Genome Informatics)

Command

Reference index script

```
minimap2 [-x preset] -d target.mmi target.fa
```

5.1

Command

Minimap2 indexing options

```
Indexing options
-k INT Minimizer k-mer length [15]

-w INT Minimizer window size [2/3 of k-mer length]. A minimizer is
the smallest k-mer in a window of w consecutive k-mers.

-H Use homopolymer-compressed (HPC) minimizers. An HPC sequence is
constructed by contracting homopolymer runs to a single base. An HPC
minimizer is a minimizer on the HPC sequence.

-I NUM Load at most NUM target bases into RAM for indexing [4G]. If
there are more than NUMbases in target.fa, minimap2 needs to read
query.fa multiple times to map it against each batch of target
sequences. NUM may be ending with k/K/m/M/g/G. NB: mapping quality is
incorrect given a multi-part index.

--idx-no-seq Don’t store target sequences in the index. It saves disk
space and memory but the index generated with this option will not
work with -a or -c. When base-level alignment is not requested, this
option is automatically applied.

-d FILE Save the minimizer index of target.fa to FILE [no dump].
Minimap2 indexing is fast. It can index the human genome in a couple
of minutes. If even shorter startup time is desired, use this option
to save the index. Indexing options are fixed in the index file. When
an index file is provided as the target sequences, options -H, -k, -
w, -I will be effectively overridden by the options stored in the
index file.

--alt FILE List of ALT contigs [null]

--alt-drop FLOAT Drop ALT hits by FLOAT fraction when ranking and
computing mapping quality [0.15]
```

6 Aligning to reference index

Command

Aligning to reference index

```
minimap2 -a -x map-ont target.mmi your_sequences.fasta > alignment.sam
```

Note

For alignment and mapping options go to
https://lh3.github.io/minimap2/minimap2.html#5

7

Command

Convert SAM output to BAM using Samtools

```
samtools view -S -b alignment.sam > alignment.bam
```

Note

The accepted file format for quantification in alignment based format for Salmon is a .BAM file. Therefore, the output from the previous step will need to be converted to a BAM in Samtools.

8 1. Quantification prior to summarization to the counts level can be done via Salmon using the alignment-based method. For ONT reads, the option –noLengthCorrection MUST be used.
2. Download the transcriptome file for your reference organism

9 Quantification using Salmon alignment based mapping

Command

Sample salmon script

```
salmon quant --ont --noLengthCorrection -p 8 -t {input.transcriptome}
-l U -a {input.bam} -o {output
```

Note

Salmon options can be found here
https://salmon.readthedocs.io/en/latest/salmon.html#using-salmon

10 Import non-normalized, raw count files to local machine and upload to Rstudio

Differential expression analysis : RStudio

11

Note

In order to run EdgeR or DEseq2 you must have biological replicates for your results to be statistically significant. Without replicates, it is not possible to differentiate between random noise and differential expression.

12 Summarization to counts level using Tximport

Command

Installation of Tximport

```
if (!require("BiocManager", quietly = TRUE))
install.packages("BiocManager") BiocManager::install("tximport")

library("tximport")
```

13 Once tximport is installed generate a sample sheet to organize your quant.sf files

14 Utilize the tximport vignette to generate script
txiimport: Import and summarize transcript-level estimates for transcript- and gene-level analysis

15 Differential expression analysis via DEseq2

Command

DESeq2 Installation command

```
if (!require("BiocManager", quietly = TRUE))
install.packages("BiocManager") BiocManager::install("DESeq2")

library("DESeq2")
```

16 Utilize the DESeq2 vignette to generate script for differential expression analysis!

DESeq2: Differential gene expression analysis based on the negative binomial distribution

Protocol references

1. “About Salmon.” *Salmon: Fast, Accurate and Bias-Aware Transcript Quantification from RNA-Seq Data*, combine-lab.github.io/salmon/about/#:~:text=The%20alignment%2Dbased%20mode%20of,wish%20to%20use%20for%20quantification. Accessed 03 Oct. 2024.

2. Belleghem, Steven Van. “Minimap2 Genome Alignment Tutorial.” *Genomics Tutorials*, 30 Mar. 2023, stevenvb12.github.io/misc/2023/03/30/Minimap2-alignment.html. Accessed 03 Oct. 2024.

3. epi2me-labs. “Epi2me-Labs/WF-Transcriptomes.” *GitHub*, github.com/epi2me-labs/wf-transcriptomes. Accessed 03 Oct. 2024.

4. “How to Align Your Data.” *EPI2ME Labs*, 29 Sept. 2023, labs.epi2me.io/how-to-align/. Accessed 03 Oct. 2024.

5. “Manual Reference Pages - Minimap2 (1).” *Manual Page - Minimap2(1)*, lh3.github.io/minimap2/minimap2.html#5. Accessed 03 Oct. 2024.

6. Michael I. Love, Simon Anders. “Analyzing RNA-Seq Data with Deseq2.” *Bioconductor*, 30 Apr. 2024, www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#input-data. Accessed 03 Oct. 2024.

7. “NanoPack: Visualizing and Processing Long Read Sequencing Data.” *Oxford Nanopore Technologies*, 5 Apr. 2024, nanoporetech.com/resource-centre/nanopack-visualizing-and-processing-long-read-sequencing-data. Accessed 03 Oct. 2024.

8. Patro, Rob. “Question for Quantification of Nanopore Reads · Issue #602 · Combine-Lab/Salmon.” *GitHub*, github.com/COMBINE-lab/salmon/issues/602. Accessed 03 Oct. 2024.

9. ProwlerForNanopore. “Prowlertrimmer/Readme.Md at Main · Prowlerfornanopore/Prowlertrimmer.” *GitHub*, github.com/ProwlerForNanopore/ProwlerTrimmer/blob/main/README.md. Accessed 03 Oct. 2024.

10. “Salmon 1.10.2 Documentation.” *Salmon*, salmon.readthedocs.io/en/latest/salmon.html#using-salmon. Accessed 03 Oct. 2024.