Jun 02, 2025

Gene re-annotations using ONT long read RNASeq data

DOI
https://dx.doi.org/10.17504/protocols.io.36wgqd5qyvk5/v1

Sophie Lemoine[1]

[1]GenomiqueENS

genomiqueENS

Sophie Lemoine
GenomiqueENS, Institut de Biologie de l'ENS (IBENS), CNRS, I...

DOI: https://dx.doi.org/10.17504/protocols.io.36wgqd5qyvk5/v1

External link: https://genomique.biologie.ens.fr/

Protocol Citation: Sophie Lemoine 2025. Gene re-annotations using ONT long read RNASeq data . **protocols.io** https://dx.doi.org/10.17504/protocols.io.36wgqd5qyvk5/v1

Protocol status: In development
We are still developing and optimizing this protocol

Created: November 14, 2024

Last Modified: June 02, 2025

Protocol Integer ID: 112118

Keywords: structural annotation, re-annotation, long-reads, read rnaseq data, rnaseq data, long reads from cdna, rna, direct rna rna, using long read, ont data, transcript boundary, long read, seq dataset, gene, using ont, seq experiment, annotation, perfect annotation, read, ont, seq

Funders Acknowledgements:
France Génomique
Grant ID: ANR-10-INBS-0009

Disclaimer

DISCLAIMER – FOR INFORMATIONAL PURPOSES ONLY; USE AT YOUR OWN RISK

The protocol content here is for informational purposes only and does not constitute legal, medical, clinical, or safety advice, or otherwise; content added to **protocols.io** is not peer reviewed and may not have undergone a formal approval of any kind. Information presented in this protocol should not substitute for independent professional judgment, advice, diagnosis, or treatment. Any action you take or refrain from taking using or relying upon the information presented here is strictly at your own risk. You agree that neither the Company nor any of the authors, contributors, administrators, or anyone else associated with **protocols.io**, can be held responsible for your use of the information contained in or linked to this protocol or any of our Sites/Apps and Services.

Abstract

This protocol offers detailed, step-by-step instructions for anyone looking to reannotate their genes using long reads from cDNA or direct RNA RNA-Seq datasets (ONT data). The goal is not to create the perfect annotation for your organisms, but to leverage your long reads to refine gene and transcript boundaries, thereby enabling more accurate counts in single-cell or RNA-Seq experiments on non-model organisms.

Before start

Retrieve your genome and annotation of interest

1 Retrieve the genome (FASTA file) and annotations (GTF or GFF file) for your organism.

Prepare your Docker environment

2 **Install Docker desktop**
https://www.docker.com/

3 **Retrieve all the necessary software from Biocontainers[1]**
Many resources are packaged as Biocontainers and can be used as Docker images or Conda packages. For an overview, visit this address: **Biocontainers Registry**.

- minimap2[2] :

```
docker pull quay.io/biocontainers/minimap2:2.28--he4a0461_3
```

- rna-bloom[3] :

```
docker pull quay.io/biocontainers/rnabloom:2.0.0--hdfd78af_0
```

- isoquant[4] :

```
docker pull quay.io/biocontainers/isoquant:3.6.1--hdfd78af_0
```

- agat[5] :

```
docker pull quay.io/biocontainers/agat:1.4.1--pl5321hdfd78af_0
```

- gffread[6] :

```
docker pull quay.io/biocontainers/gffread:0.12.7--hdcf5f25_4
```

- samtools [7]

Retrieve Restrander from Docker hub :

If what you need is not available as a Biocontainer, you can search the Docker Hub image library (**https://hub.docker.com/**) and even upload your own if you wish to share it.

- restrander[8] :

```
docker pull genomicpariscentre/restrander
```

Ensure you have everything doing :

```
docker images
```

Tips :
Be sure to launch your Docker images in screen sessions so that you can detach each of them from your terminal.

```
screen # open a screen session

[CTRL]+[a][d] # Type this to close this screen session

screen -r # List the different screen sessions detached

screen -r id # Attach id session to your terminal
```

Prepare your ONT reads

4

Re-strand your reads using Restrander
Unless you already know the version of the sequencing kit used to generate your data, contact the sequencing facility you worked with to obtain the reference. My example

pertains to the PCB111 version, but it could also be PCB114 or another version. This will determine the configuration JSON file to use.
Here is the PCB111.json file: **PCB111.json**

```
{
    "name": "PCB111",
    "description": "Looks for the standard TSO (SSP) and RTP (VNP)
used in PCB111 chemistry.",
    "pipeline": [
        {
            "type": "poly",
            "tail-length": 12,
            "search-size": 200
        },
        {
            "type": "primer",
            "tso": "TTTCTGTTGGTGCTGATATTGCTTT",
            "rtp": "CTTGCCTGTCGCTCTATCTTCAGAGGAG",
            "report-artefacts": true
        }
    ],
    "silent": false,
    "exclude-unknowns": true,
    "error-rate": 0.25
}
```

If you have your own primers, you can create a custom JSON file and reference it in your command line.
For more detailed assistance, refer to the restrander vignette: https://github.com/jakob-schuster/restrander-vignette.

Let's assume :

- your analysis directory is /location_of_your_analyses/
- your FASTQ file location is /location_of_your_fastq_files/

Now, launch the restrander image:

```
docker run -u GID:UID \
 -v /location_of_your_analyses/:/data \
 -v /location_of_your_fastq_files/:/fastq \
 -t -i --rm genomicpariscentre/restrander:1.0.0 bash
```

Within your Docker environment :

- your analyses directory is mounted as /data
- your FASTQ file directory is mounted as /fastq

```
cd /data
mkdir restrander

# Launch restrander

/usr/local/restrander/restrander /fastq/SQK-PCB111-
24_barcode11.fastq.gz \
 /data/restrander/SQK-PCB111-
24_barcode11_restranded_PCB111.fastq.gz \
 /usr/local/restrander/config/PCB111.json \
 > output-stat_barcode11_restranded_PCB111.json
```

You now have three files in your output directory :

- SQK-PCB111-24_barcode11_restranded.fastq.gz : your successfully restranded reads
- SQK-PCB111-24_barcode11_restranded-unknowns.fastq.gz
- output-stat_SQK-PCB111-24_barcode11_restranded.txt : a log file structured as follows


```
{
    "stats": {
        "artefactStats": {
            "RTP-RTP": 6908,
            "TSO-TSO": 52611,
            "no artefact": 10218243
        },
        "strandStats": {
            "+": 4868592,
            "-": 3582672,
            "?": 1826498
        },
        "totalReads": 10277762
    }
}
```


RNA-Bloom (reference-free transcriptome assembly for short and long reads Topics)

5 **Launch the RNA-Bloom Docker image and execute RNA-Bloom**

```
docker run -u GID:UID  \
 -v /location_of_your_analyses/:/data \
 -t -i --rm -e HOME=/data/ \
 genomicpariscentre/rnabloom:v2.0.1 bash

# Go to your analysis directory
cd /data
mkdir rnabloom
# Execute RNA-Bloom on your restranded reads
# Output in rnabloom directory
java -jar /usr/local/RNA-Bloom_v2.0.1/RNA-Bloom.jar \
 -long /data/restrander/SQK-PCB111-
24_barcode11_restranded.fastq.gz \
 -stranded \
 -t 12 \
 -outdir /data/rnabloom/
```

If you want to assemble long-read sequencing data with short-read polishing, use the following options:
-ser is for reverse reads
-sef would be for forward reads

```
docker run -u GID:UID  \
 -v /location_of_your_analyses/:/data \
 -v /location_of_your_illumina_data/:/illumina \
 -t -i --rm -e HOME=/data/ \
 genomicpariscentre/rnabloom:v2.0.1 bash

# Go to your analysis directory
cd /data
mkdir rnabloom
# Execute RNA-Bloom on your restranded reads
# Output in rnabloom directory
java -jar /usr/local/RNA-Bloom_v2.0.1/RNA-Bloom.jar \
 -long /data/restrander/SQK-PCB111-
24_barcode11_restranded.fastq.gz \
 -stranded \
 -ser /illumina/reads_*.fq
 -t 12 \
 -outdir /data/rnabloom/
```

Porechop is recommended in the documentation, but adapters can now be trimmed during the demultiplexing step of cDNA sequencing. For more details, refer to the RNA-Bloom GitHub page (https://github.com/bcgsc/RNA-Bloom).
Within your output directory, you will find numerous intermediate files, including the rnabloom.transcripts.fa file.

6 **Align the RNA-Bloom transcripts to the genome (minimap2)**

Since RNA-Bloom employs a reference-free approach, it provides a transcript FASTA file. To proceed further, you need to map it to your reference genome.
Launch your Minimap2 instance and map your file to the reference file:

```
docker run -u GID:UID  \
 -v /location_of_your_analyses/:/data \
 -v /location_of_genome_fasta_file/:/annot \
 -t -i --rm \
 quay.io/biocontainers/minimap2:2.28--he4a0461_3 bash

cd /data/

minimap2 -G 20000 \ # Choose the right max intron length
 -ax splice \
 -uf \ # Your reads are forward
 -k14 \
 /annot/genome.fa \
 /data/rnabloom/rnabloom.transcripts.fa \
 > /data/rnabloom/rnabloom.transcripts.sam
```

7 **Convert your RNA-Bloom SAM file into a BED12 file (paftools in Minimap2)**

Paftools is a very convenient utility tool contained in minimap2 :

```
Usage: paftools.js  [arguments]
Commands:
  view       convert PAF to BLAST-like (for eyeballing) or MAF
  splice2bed convert spliced alignment in PAF/SAM to BED12
  sam2paf    convert SAM to PAF
  delta2paf  convert MUMmer's delta to PAF
  gff2bed    convert GTF/GFF3 to BED12
  gff2junc   convert GFF3 to junction BED
  longcs2seq convert long-cs PAF to sequences
  stat       collect basic mapping information in PAF/SAM
  asmstat    collect basic assembly information
  asmgene    evaluate gene completeness
  misjoin    evaluate large-scale misjoins
  liftover   simplistic liftOver
  call       call variants from asm-to-ref alignment with the cs
tag
  bedcov     compute the number of bases covered
  vcfstat    VCF statistics
  sveval     compare two SV callsets in VCF
  version   print paftools.js version
  mapeval    evaluate mapping accuracy using mason2/PBSIM-
simulated FASTQ
  pafcmp     compare two PAF files
  mason2fq   convert mason2-simulated SAM to FASTQ
  pbsim2fq   convert PBSIM-simulated MAF to FASTQ
  junceval   evaluate splice junction consistency with known
annotations
  exoneval   evaluate exon-level consistency with known
annotations
  ov-eval    evaluate read overlap sensitivity using read-to-ref
mapping
```

No need to leave your minimap2 docker image, use paftools to convert your SAM file into a BED12 file as following :

```
paftools.js splice2bed /data/rnabloom/rnabloom.transcripts.sam >
/data/rnabloom.transcripts.bed
```

8 Convert your BED12 file into a GFF file and then in GTF file (AGAT)

We now need to convert the BED12 file into a GTF file. File conversion is a significant concern, as we want to avoid losing any information during the process. After conducting various tests, we can affirm that AGAT is an excellent toolkit for this purpose (**https://nbisweden.github.io/AGAT/**).

```
docker run -u GID:UID  \
 -v /location_of_your_analyses/:/data \
 -v /location_of_genome_fasta_file/:/annot \
 -t -i --rm \
 quay.io/biocontainers/agat:1.4.1--pl5321hdfd78af_0 bash

cd /data/rnabloom
agat_convert_bed2gff.pl --bed rnabloom.transcripts.bed -o
rnabloom.transcripts.gff #bed12 to gff
agat_convert_sp_gff2gtf.pl --gff rnabloom.transcripts.gff -o
rnabloom.transcripts.gtf #gff to gtf
```

Align your restranded reads to the genome

9 **Align each restranded read from your FASTQ file to the genome**

```
docker run -u GID:UID  \
 -v /location_of_your_analyses/:/data \
 -v /location_of_genome_fasta_file/:/annot \
 -t -i --rm \
 quay.io/biocontainers/minimap2:2.28--he4a0461_3 bash

cd /data/

minimap2 -G 20000 \ # Choose the right max intron length, ideally
the same as what you chose for rnabloom
 -ax splice \
 --eqx \
 --secondary=no \
 /annot/genome.fa \
 /data/restrander/sample1_restranded.fastq.gz  \
 > /data/samfiles/sample1.sam
```

Convert your SAM files into BAM files

```
docker run -u GID:UID  \
 -v /location_of_your_analyses/:/data \
 -v /location_of_genome_fasta_file/:/annot \
 -t -i --rm \
 biocontainers/samtools bash

cd /data/
samtools faidx /annot/genome.fa #Index your genome
samtools view -bt /annot/genome.fa.fai /data/samfiles/sample1.sam
> /data/bamfiles/sample1.bam #Convert your sam into bam
#Sort and index your bam file
samtools sort /data/bamfiles/sample1.bam -o
/data/bamfiles/sample1_sorted
samtools index /data/bamfiles/sample1_sorted.bam
/data/bamfiles/sample1_sorted.bam.bai
```

IsoQuant (Transcript discovery and quantification with long RNA reads - Nanopore and PacBio)

10 **Describe your samples in a yaml file** (See IsoQuant web pages for a more detailed description of the yaml file)

```
[
  data format: "bam",
  {
    name: "Reannot",
    long read files: [
      "/data/bamfiles/sample1_sorted.bam",
      "/data/bamfiles/sample2_sorted.bam"
    ],
    labels: [
      "techRep1",
      "techRep2"
    ]
  }
]
```

Launch IsoQuant

```
docker run -u GID:UID  -v /location_of_your_analyses/:/data \
 -v /location_of_genome_and_annotation_file/:/annot  \
quay.io/biocontainers/isoquant:3.6.1--hdfd78af_0  bash

isoquant.py \
 --reference /annot/genomes/genome.fa \
 --data_type nanopore \
 --stranded forward \
 --clean_start \
 --model_construction_strategy default_ont \
 --yaml /data/isoquant_3.6.1/samples.yaml \
 --output /data/isoquant_3.6.1
```

You can include gene annotation in the command line, which is mandatory if you aim to quantify genes or transcripts. If you intend to reannotate your transcripts, my experience suggests that introducing it later yields better results.

```
--genedb /annot/gtf/annot.gtf
```

IsoQuant has well documented web pages. Feel free to check each option to better suit your needs (**https://ablab.github.io/IsoQuant/**)
IsoQuant outputs a gtf file named Reannot.transcript_models.gtf (/location_of_your_analyses/isoquant_3.6.1/Reannot/Reannot.transcript_models.gtf)

Consensus

11 I do not recommend introducing the official annotation at this stage. RNA-Bloom generates many transcripts, and I plan to use the -K option in Gffread to filter them out. Since our transcripts may be longer than those in the official annotation, they could be omitted in our consensus file. The official annotation, which is useful for obtaining gene names and other metadata, should be utilized after this step.

Construct a merged GTF file using AGAT based on the IsoQuant and RNA-Bloom files

```
docker run --rm -it \
-u GID:UID \
 -v /location_of_your_analyses/:/data \
 quay.io/biocontainers/agat:1.4.1--pl5321hdfd78af_0 bash

agat_sp_complement_annotations.pl \
 --ref /data/isoquant_3.6.1/Reannot/Reannot.transcript_models.gtf
\
 --add /data/rnabloom/rnabloom.transcripts.gtf \
 -o /data/consensus/isoquant_rnabloom.gff
```

Construct a transcript consensus between the merged GTF files (RNA-Bloom and IsoQuant) using Gffread

```
docker run --rm -it \
 -u GID:UID \
 -v /location_of_your_analyses/:/data \
 quay.io/biocontainers/gffread:0.12.7--hdcf5f25_4 bash

gffread -g /annot/genomes/genome.fa \
 -o /data/consensus/consensus_gffread_MKYZ.gff \
 -M -K -Y -Z --keep-comments \ # To be customized, options are not
always so clear
 /data/consensus/isoquant_rnabloom.gff
```

12 **Fix the consensus gtf output**

1. *;locus=* has to be replaced by *;Parent=* in the transcript attribute field
2. *locus* has to be replaced by *gene* in the feature field

13 **Name the genes according to the official annotation**

Work in progress to provide a clean and reproducible script. For now, it remains species dependant.

Nextflow pipeline under development

14 This protocol is being developed by Salomé Brunon and Laurent Jourdren within a nextflow pipeline called Egzotek[9].

Protocol references

1- Gruening, B., Sallou, O., Moreno, P., da Veiga Leprevost, F., Ménager, H., Søndergaard, D., Röst, H., Sachsenberg, T., O'Connor, B., Madeira, F. and Del Angel, V.D., BioContainers Community, Perez-Riverol Y. 2018. **Recommendations for the packaging and containerizing of bioinformatics software**. F1000Research, 7
2- Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, **34**:3094-3100. **doi:10.1093/bioinformatics/bty191**
3- Ka Ming Nip, Saber Hafezqorani, Kristina K. Gagalova, Readman Chiu, Chen Yang, René L. Warren, and Inanc Birol. Reference-free assembly of long-read transcriptome sequencing data with RNA-Bloom2. Nature Communications. 2023 May 22;14(1):2940. doi: 10.1038/s41467-023-38553-y
4- Prjibelski, A.D., Mikheenko, A., Joglekar, A. *et al.* Accurate isoform discovery with IsoQuant using long reads. *Nat Biotechnol* **41**, 915–918 (2023). https://doi.org/10.1038/s41587-022-01565-y
5- Dainat J. 2022. Another Gtf/Gff Analysis Toolkit (AGAT): Resolve interoperability issues and accomplish more with your annotations. Plant and Animal Genome XXIX Conference. https://github.com/NBISweden/AGAT.
6- Pertea G and Pertea M. GFF Utilities: GffRead and GffCompare [version 2; peer review: 3 approved]. F1000Research 2020, 9:304 (https://doi.org/10.12688/f1000research.23297.2)
7- 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, The Sequence alignment/map (SAM) format and SAMtools, *Bioinformatics* (2009) 25(16) 2078-9
8- Schuster J, Ritchie ME, Gouil Q. Restrander: rapid orientation and artefact removal for long-read cDNA data. NAR Genom Bioinform. 2023 Dec 23;5(4):lqad108. doi: 10.1093/nargab/lqad108. PMID: 38143957; PMCID: PMC10748469.
9- Brunon Salomé, Jourdren Laurent, https://github.com/GenomiqueENS/egzotek

Acknowledgements

This work was supported by the France Génomique national infrastructure, funded as part of the "Investissements d'Avenir" program managed by the Agence Nationale de la Recherche (contract ANR-10-INBS-0009)