

Local ORF Detector Manual (LOD)

Table of Contents

- Introduction
 - About this pipeline
 - Workflow
- Packages
 - Samtools
 - HiSAT2
 - STAR
 - StringTie
 - RiboCode
- Package Installation
- Running Pipeline
 - Required Input Files
 - Usage
 - Command Line
 - Parameters
- Getting Started with the Pipeline
 - HiSAT2
 - STAR
 - StringTie
 - preprocessGTF
 - ChangeName
 - STAR
 - RiboCode
- Pitfalls and Limitation

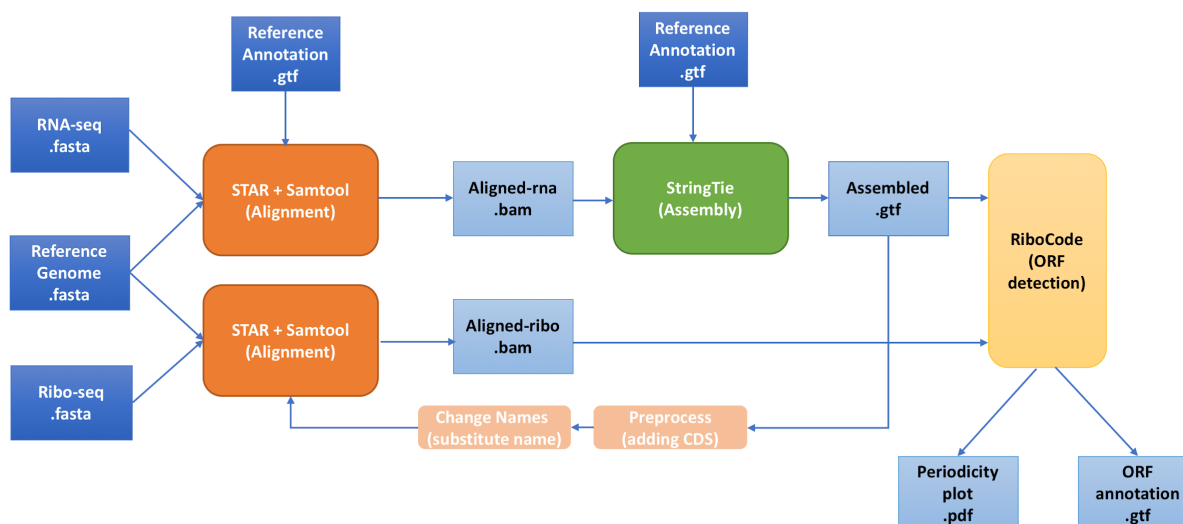
About this Pipeline

This new pipeline of annotating open reading frames (ORFs) using both RNA-seq and Ribo-seq data integrates multiple bioinformatics tools. *RiboCode* is a newly-invented public tool that can make use of Ribo-seq data to identify ORFs. The reference .gtf file that RiboCode requires normally comes from the public database and therefore loses the specificity in terms of annotating samples from a specific cell type or condition. Here we introduce a solution of making a tailored experimental reference transcript based on the RNA-seq data that are gathered from the same experiment. We use *HiSAT2* for RNA-seq alignment and *StringTie* for transcript assembly to generate the experimental reference .gtf file. Then the Ribo-seq reads are aligned with *STAR* and the resulting alignment files are fed into RiboCode together with our newly assembled transcript to call ORFs.

```
graph LR
    RA1[Reference Annotation .gtf] --> EF[Exon file]
    RA1 --> SSF[Splice site file]
    RNAseq[RNA-seq .fasta] --> HISAT2[HiSAT2 + Samtool Alignment]
    RefGenome[Reference Genome .fasta] --> HISAT2
    RefGenome --> STAR[STAR + Samtool Alignment]
    Riboseq[Ribo-seq .fasta] --> STAR
    EF --> HISAT2
    SSF --> HISAT2
    HISAT2 --> AlignedRNA[Aligned-rna .bam]
    AlignedRNA --> StringTie[StringTie Assembly]
    StringTie --> Assembled[Assembled .gtf]
    STAR --> AlignedRibo[Aligned-ribo .bam]
    Assembled --> RiboCode[RiboCode ORF detection]
    AlignedRibo --> RiboCode
    RiboCode --> Preprocess[Preprocess adding CDS]
    Preprocess --> ChangeNames[Change Names substitute name]
    ChangeNames --> STAR
    RiboCode --> Periodicity[Periodicity plot .pdf]
    RiboCode --> ORFAnnotation[ORF annotation .gtf]
```

The flowchart illustrates the RiboCode pipeline for ORF detection. It starts with a **Reference Annotation .gtf** file, which is used to generate an **Exon file** and a **Splice site file**. The **RNA-seq .fasta** file is aligned using **HiSAT2 + Samtool Alignment**, while the **Reference Genome .fasta** and **Ribo-seq .fasta** files are aligned using **STAR + Samtool Alignment**. The **Exon file** and **Splice site file** are also inputs to the **HiSAT2 + Samtool Alignment** step. The **HiSAT2 + Samtool Alignment** step produces an **Aligned-rna .bam** file, which is then processed by **StringTie Assembly** to generate an **Assembled .gtf** file. The **STAR + Samtool Alignment** step produces an **Aligned-ribo .bam** file. Both the **Assembled .gtf** file and the **Aligned-ribo .bam** file are inputs to the **RiboCode (ORF detection)** step. The **RiboCode (ORF detection)** step produces a **Periodicity plot .pdf** and an **ORF annotation .gtf** file. The **ORF annotation .gtf** file is then processed by **Preprocess (adding CDS)** and **Change Names (substitute name)** steps, which feed back into the **STAR + Samtool Alignment** step.

Alternatively, if the users do not have enough memory to run HiSAT2, user will eventually be able to use STAR as a backup aligner to carry out the first round of RNA-seq alignment too. However, at the moment, this has not been implemented yet.



Packages

Note: Our pipeline incorporates all the following packages in binary format except for RiboCode. User could only try to install RiboCode manually to set up all the environment needed. For parameters please refer to Running Pipeline-Parameters section.

Samtools

Sequence Alignment/Map tools(Samtools) is a package that takes .sam and .bam files as inputs to implement utilizes for post-processing alignment. The most frequently used functions of samtools are viewing the aligned files, sorting the aligned files (which is required before assembly and many other following operations), and converting between .sam files and .bam files.

For more details, please refer to:

<https://www.ncbi.nlm.nih.gov/pubmed/19505943>

Download and build the environment of Samtools from:

<https://github.com/samtools/samtools>

HiSAT2

Hierarchical indexing for spliced alignment for transcripts (HiSAT2) is an aligner that maps sequenced reads to the reference genome considering the alternative splicing.

For more details, please refer to:

<https://www.ncbi.nlm.nih.gov/pubmed/25751142>

Download and build the environment of HiSAT2 from:

<https://ccb.jhu.edu/software/hisat2/manual.shtml#building-from-source>

STAR

Spliced Transcripts Alignment to a Reference (STAR) is an algorithm designed to align high-throughput RNA-seq data to a reference file using a strategy for spliced alignments.

For more detail, please refer to:

<https://www.ncbi.nlm.nih.gov/pubmed/23104886>

Download and build the environment of STAR from:

<https://github.com/alexdobin/STAR>

StringTie

StringTie is an assembler of RNA-Seq alignments into potential transcripts. It takes the alignments of raw reads as input and can be run in either reference-guide or *de novo* mode. In our pipeline, transcripts are assembled only in a reference-guide mode. By doing so, we sacrifice the ability to identify new transcripts to gain benefits in specificity.

For more detail, please refer to:

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4643835/>

Download and build the environment of StringTie from:

<https://github.com/gpertea/stringtie>

RiboCode

RiboCode, a newly developed statistical method, used for gene annotation and characterization. RiboCode takes aligned-ribo.bam and the assemble .gtf as inputs and aims to identify translated ORFs using the ribosome-profiling data by outputting the periodicity plot and the ORF annotation .gtf files.

For more detail, please refer to:

<https://www.ncbi.nlm.nih.gov/pubmed/29538776>

Download and build the environment of riboCode from:

<https://github.com/xryanglab/RiboCode>

Package installation

The only package the users need to install manually is RiboCode. To do so, pip must have write privileges for folders in the path, python-devel must be installed, and pip must be installed

```
pip install RiboCode
```

For full documentation please see: <https://github.com/xryanglab/RiboCode>

Running Pipeline

Required Input Files:

All the reference files can be downloaded from e!Ensemble:

<http://useast.ensembl.org/info/data/ftp/index.html>

All the experimental data can be generated either by the users or downloaded from the public database like NCBI: <https://www.ncbi.nlm.nih.gov/>

1. Reference genome fasta file: Serves as reference genome file for STAR and RiboCode.
2. Reference annotation gtf file: Serves as reference annotation file for StringTie. Our newly assembled transcript reference gtf file is built in a reference-guided mode.
3. RNA-seq fasta file: experimental RNA profiling to build our own transcripts.
4. Ribo-seq fasta file: experimental Ribosome profiling to generate ORF annotation.

Usage:

The LOD.py script can NOT be moved from the project folder. The relative path to the binaries is used.

The user can use LOD in two ways. The full path to the executable can be used, or the user can create a sym-link in their path for convenience. By creating a sym-link, LOD.py can be called from anywhere without having to specify the full path.

```
ln -s /path/to/LOD.py /usr/local/bin/
```

Command Line:

Example:

```
python LOD.py --refFa <Mus_musculus.GRCm38.75.dna.primary_assembly.fa>  
--refGTF <Mus_musculus.GRCm38.75.gtf> --RNASeq <RnaSeq.fa> --RiboSeq  
<RiboSeq.fa> --f 0.99 --n 25 --i
```

Parameters:

Required:

- refFa: Reference genome fasta file.
- refGTF: Reference annotation gtf file
- RNASeq: RNA-seq fasta file
- RiboSeq: Ribo-seq fasta file

Optional:

--n: numThreads. If using a large number of threads (~20-30) be aware of the max number of files your system allows. You may need to adjust this to avoid harddrive bomb. <https://github.com/alexdobin/STAR/issues/269>

--f: when assembling transcripts using StringTie, the user can set the minimum isoform abundance of the predicted transcripts as a fraction of the most abundant transcript assembled at a given locus <0.0-0.1>. The default fraction is set at a relatively high level to eliminate transcript isoforms with low confidence. Default: 0.9

--i: when running the pipeline, we call HiSAT2 by default. Add the file name if the user wants to provide their own indexed file.

Getting Started with the Pipeline

For your convenience, we put all the intermediate files in the ./intermediate/ folder, only the final .gtf file, some Log files we haven't gotten around to remove yet, and the PDF file would be in the current directory.

HiSAT2:

HiSAT2 will generate .sam file from the RNA-seq data, reference genome, and the reference annotation file in the ./intermediate/ folder. It is the alignment result of RNA-seq reads to the reference genome. Samtools is then used to make a .bam file from the alignment result.

STAR (Alternative):

The first time through STAR will generate .bam file which would be in the ./intermediate/ folder for aligning the RNA-seq reads to the reference genome. Samtools is then used to make a .bam file. The second time through STAR after StringTie takes the assemble .gtf file as input and generates a .bam file in the ./intermediate/ folder that aligned the Ribo-seq data to the experimental genome.

STAR will first index the reference genome using the reference .gtf file. To do this, we will create a ./intermediate/ folder to hold these index files. Then in the next step, STAR aligns the RNA-seq reads to the indexed reference genome, creating a .sam file (shown below). The .sam file is then converted to a .bam file and sorted using samtools.

Sample output:

```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001 99 ref 7 30 8M2I4M1D3M = 37 39 TTAGATAAAGGATACTG *
r002 0 ref 9 30 3S6M1P1I4M * 0 0 AAAAGATAAGGATA *
r003 0 ref 9 30 5S6M * 0 0 GCCTAAGCTAA * SA:Z:ref,29,-,6H5M,17,0;
r004 0 ref 16 30 6M14N5M * 0 0 ATAGCTTCAGC *
r003 2064 ref 29 17 6H5M * 0 0 TAGGC * SA:Z:ref,9,+,5S6M,30,1;
r001 147 ref 37 30 9M = 7 -39 CAGCGGCAT * NM:i:1
```

In a .sam file, each line stands for a linear segment that is aligned. The standard format of a .sam file include 11 fields for each entry. These fields are mandatory and should always appear in the same order. If the information is missing, the default values can be imputed with either '0' or '*' depending on the specific field. The following table shows an overview of the 11 fields in the SAM format:

Col	Field	Type	Regex/Range	Brief description
1	QNAME	String	[!~?A~]{1,254}	Query template NAME
2	FLAG	Int	[0,2 ¹⁶ -1]	bitwise FLAG
3	RNAME	String	* [!-()+-<>-~] [!-~]*	Reference sequence NAME
4	POS	Int	[0,2 ³¹ -1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2 ⁸ -1]	MAPping Quality
6	CIGAR	String	* ([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	* = [!-()+-<>-~] [!-~]*	Ref. name of the mate/next read
8	PNEXT	Int	[0,2 ³¹ -1]	Position of the mate/next read
9	TLEN	Int	[-2 ³¹ +1,2 ³¹ -1]	observed Template LENgth
10	SEQ	String	* [A-Za-z=.] +	segment SEQUENCE
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

Please refer to samtool manual for more information:

<https://samtools.github.io/hts-specs/SAMv1.pdf>

StringTie:

StringTie would generate .gtf file from the alignment results provided in .bam format, the assembly is carried out in a guided mode, therefore another reference gtf file is also required. The resulting .gtf file would be in the ./outputs/ folder. The standard format of a .gtf file contains the following 9 fields in the exact order: seqname, source, feature, start_position, end_position, score, strand, frame and attributes as a string including information such as gene_id, transcript_id, exon_number, reference_id, ref_gene_id, ref_gene_name, coverage, FPKM, and TPM.

Sample output:

```

1 StringTie transcript 3102016 3102125 1000 + . gene_id "ENSMUSG00000064842"; transcript_id "ENSMUST00000082908"; ref_gene_name "Gm26206"; cov "0.727
1 StringTie exon 3102016 3102125 1000 + . gene_id "ENSMUSG00000064842"; transcript_id "ENSMUST00000082908"; exon_number "1"; ref_gene_name "Gm26206
1 StringTie transcript 4529017 4529123 1000 + . gene_id "ENSMUSG00000096126"; transcript_id "ENSMUST00000180019"; ref_gene_name "Gm22307"; cov "0.053
1 StringTie exon 4529017 4529123 1000 + . gene_id "ENSMUSG00000096126"; transcript_id "ENSMUST00000180019"; exon_number "1"; ref_gene_name "Gm22307
1 protein_coding transcript 4491713 4496363 . - . gene_id "ENSMUSG00000025902"; transcript_id "ENSMUST00000116652"; ref_gene_name "Sox17"; cov "0.0"; F
1 protein_coding exon 4491713 4492668 . - . gene_id "ENSMUSG00000025902"; transcript_id "ENSMUST00000116652"; exon_number "1"; ref_gene_name "Sox17";
1 protein_coding exon 4493100 4493490 . - . gene_id "ENSMUSG00000025902"; transcript_id "ENSMUST00000116652"; exon_number "2"; ref_gene_name "Sox17";
1 protein_coding exon 4493772 4493863 . - . gene_id "ENSMUSG00000025902"; transcript_id "ENSMUST00000116652"; exon_number "3"; ref_gene_name "Sox17";

```

Please refer to StringTie manual for more information:

<https://ccb.jhu.edu/software/stringtie/index.shtml?t=manual>

preprocessGTF:

A python script that adds the CDS annotation and the start/stop codon annotations in the reference to the newly assembled gtf file after StringTie. The entries are added whenever there's a matching transcript found in the reference gtf file. The CDS and start/stop codon information are required in order to build the periodicity when trying RiboCode.

ChangeName:

A python script that changes the gene_id and the transcript_id to the reference_gene_id and reference_id respectively whenever there's a matching transcript/exon found in the reference .gtf file. It keeps the newly discovered transcripts with the names that the StringTie randomly assigned, possibly starting with 'STRG'.

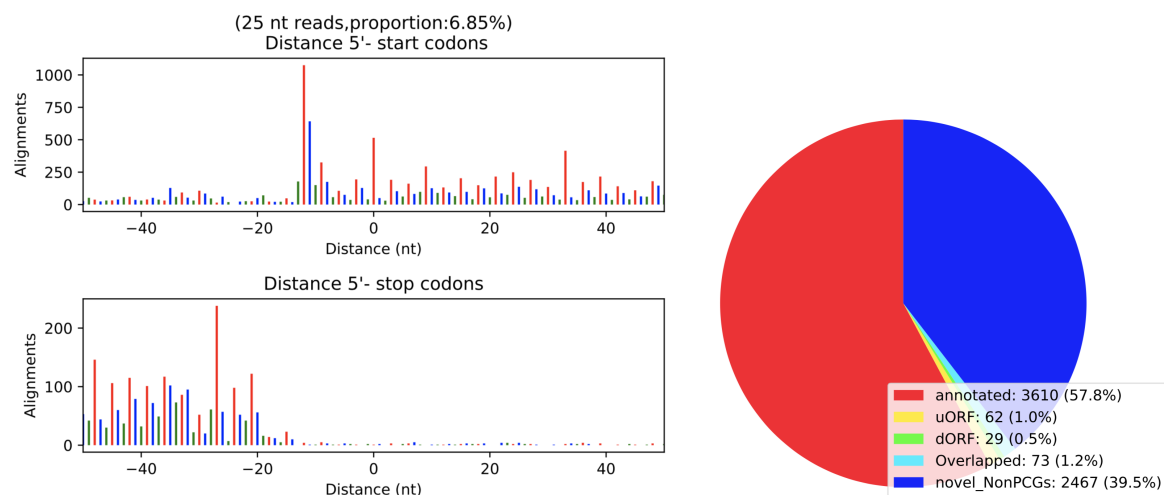
STAR:

STAR will first index the reference genome using the newly assembled experimental .gtf file. To do this, we will create a directory to hold these index file. Then STAR aligns the Ribo-seq reads to the indexed experimental genome, creating a sam file. The sam file is then converted to a bam file and sorted using samtools.

RiboCode:

RiboCode would first generate a <RiboCode_annot> folder in the ./intermediate/ folder that contains <transcript_cds.txt>, <transcripts_sequence.fa>, and <transcripts.pickle> when running 'prepare_transcripts'. These are only intermediate files for following steps, ignore them if you only want the final results. The second step, 'metaplots', would output a PDF file in current folder which plots the aggregate profiles of the distance from the 5'-end of reads to the annotated start codons (or stop codons). The last step is running 'ribocode' and this would generate a <RiboCode_ORFs_result.gtf> file in current folder which contains the predicted ORFs.

Sample output:



Pitfalls and Limitations:

Currently the pipeline requires a large amount of computational requirements. When Hisat is provided a .gtf file for indexing, it can take 200GB of RAM to index. When aligning reads to the indexed genome, if the number of threads is too high, thousands of temporary files can be written. Most computers have a limit on the number of open files at a time to prevent fork/recursion bombs. This will cause the program to crash.

The nature of the novel ORFs detected has not been evaluated. It is entirely possible that many of these novel ORFs have a similar region in the reference annotations and StringTie failed to create them during the assembly step.