Database of eCLIP Assays for Sequence Similarity Searches

The DEA

---User Manual

Team 2: Jianpeng Yu, Neel Mittal, Rishi Verma, Yuying Bian, Zhiyun Gong

Table of Content

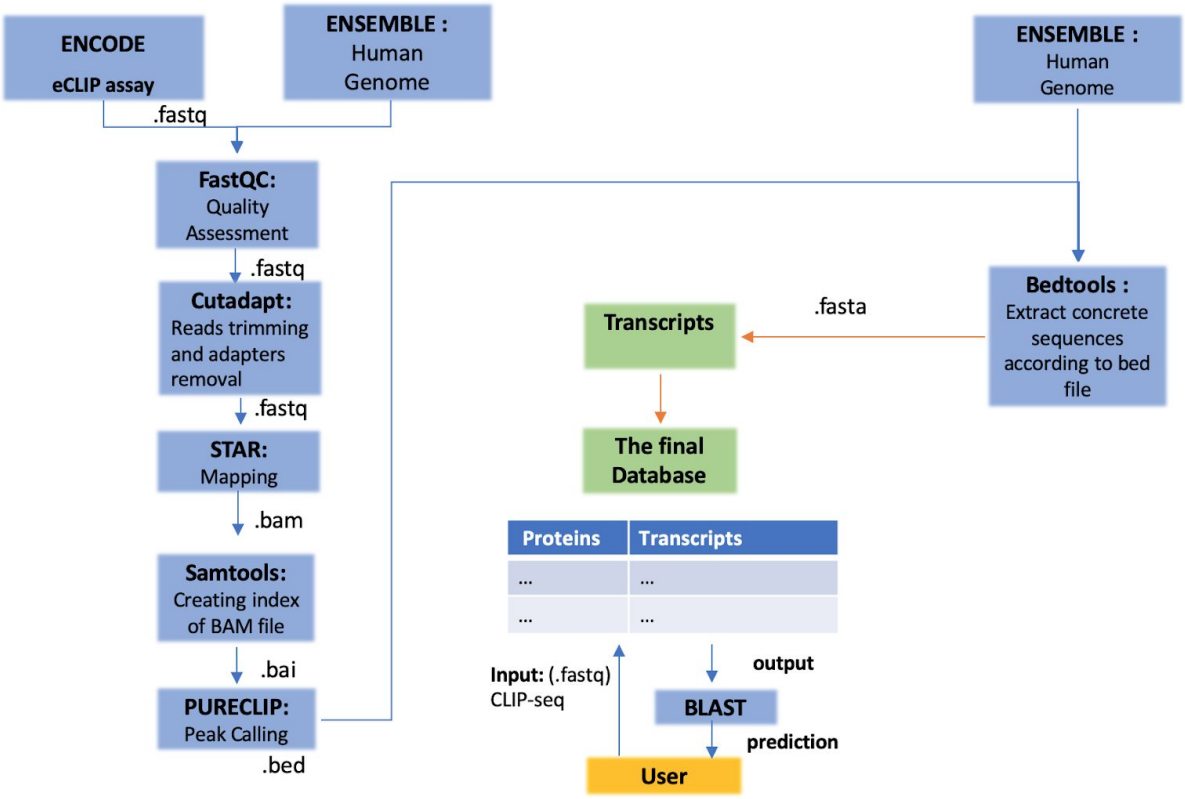
Introduction

In our project, we designed a pipeline and database that integrates computational tools for biologists to find potential RBP (RNA-binding proteins) binding peak sequences. Through our pipeline a scientist can input their raw CLIP-seq experimental data and acquire the processed data BLAST searching and comparisons against a database of known RBP sequences.

Our database is built from the processed eCLIP data on ENCODE, where we collected all CLIP bed files. When researchers have new CLIP-seq fastq files from their own experiment, they can use our pipeline and database to find out which RBPs in ENCODE share similar binding sites to their own discovered RBP. The database could allow for peak sequence comparison, highlighting regions of the genome that multiple proteins may bind to and predicting protein function.

Pipeline and Workflow

Steps:



Database:
The database containing sequences of all known RBP sites was curated using blast, specifically the **makeblastdb** command. The sequences were obtained from the .bed files of every assay on ENCODE, and the '**bedtools getfasta**' command was used to extract significant sequences. The final database contains 1,285,014 sequences of known RNA binding protein sites, and when queried along with an E-value will return significant results in blast output format.

Materials and Methods

1. eCLIP data:

Human eCLIP assays, processed data: narrow peak bed files from ENCODE. For more information please refer to:
https://www.encodeproject.org/matrix/?type=Experiment&status=released&assay_slims=RNA+binding&award.project=ENCODE&assay_title=eCLIP

2. Masked reference Genome:

Shorter reads would map to repetitive parts of the genome more frequently and cause a bias when counting reads, so the reference genome used for alignment was masked due to the short length of the CLIP reads. In DEA directory, we provided a default masked reference Genome of hg38 version.

3. FastQC (0.11.3):

FastQC is used for raw sequence data quality assessment and the results display potential problems in the sequence data, which allows users to make adjustments before analysis. FastQC accepts data from BAM, SAM and FastQ files, and exports HTML result reports. In this project's pipeline, we use FastQ files as inputs and store the output results to FastqcOutput folder.

More details and download, please refer to:
https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
https://github.com/s-andrews/FastQC

4. Cutadapt (1.5):

In the raw sequence data, there may be unwanted sequences such as adapter sequences, primers, or poly-A tails. Cutadapt trims input sequence data and removes such unwanted sequences. The input and output data are both fastq

files. Because the eCLIP experiments do not use the usual illumina adaptors in sequencing, we followed Yeo lab eCLIP-seq Processing Pipeline to remove corresponding adaptors.

For more information, please refer to:
https://github.com/marcelm/cutadapt

5. STAR (2.7.0e):

To identify where RBP binding sequences are produced from in the DNA, CLIP reads need to be aligned to the human genome. Spliced Transcripts Alignment to a Reference (STAR) is an alignment algorithm for exactly this purpose. STAR takes fastqc files as input and then outputs bam files. There are two steps for implementation:

(1) Create reference genome index
(2) Align sequences with STAR

For more information, please refer to:
https://github.com/alexdobin/STAR

6. Samtools:

There are a set of tools in Samtools and in this project we mainly used it to create new index files of BAM files, which can make the peak calling process work more efficiently. Users need to input BAM file and the output file format is BAI.

For more information, please refer to:
https://github.com/samtools/samtools

7. PureCLIP:

PureCLIP is a method based on the hidden Markov Model and used to capture target-specific protein–RNA interaction sites from iCLIP/eCLIP-seq data. In this project, we choose to use PureCLIP to do peak calling. Users need to import BAI files from Samtools results and they will get BED files as output.

For more information, please refer to:
https://github.com/skrakau/PureCLIP

8. Bedtools (v2.29.2):

Bedtools is a toolset for multiple genomic data analysis tasks. Here we use bedtools to extract concrete sequences using bed files (the result from CLIPper) and a human reference genome. The output will be used for motif scanning and discovery.

For more information, please refer to:
https://github.com/arq5x/bedtools

9. BLAST:

We built a database by using BLAST, which contains almost all human eCLIP assays narrow peak files from ENCODE. When users have already run through 1-8 tools, they can use the running result as input to search in the database. Finally, they can attain the blast result and know which proteins in our database have the most similar sequences with the one that the user is interested in.

Installation

For installation details, please refer to this github page:
https://github.com/team2DEA/CLIP-Seq-pipeline

Please run the requirements bash script as it will load all the previously mentioned software. Make sure you add your DEA directory to the path variable.

The commands are following:

```
$source requirements
$export PATH=$PATH:$(pwd)
$source PATH
```

Running the Pipeline

I. **Input files:**

Required:

1. Paired-end CLIP-Seq results in .fastq format

Optional:

Note: The 1st and 2nd optional parameters need to be specified together

1. The reference genome/transcriptome sequence sequence (in .fasta format) OR the folder containing the STAR index of the reference

genome/transcriptome that the user wants to map the reads against (default: hg38 with repetitive elements masked)
2. The annotation of the reference genome/transcriptome in .gtf format (default: hg38.gtf)
3. Alignment result file in .bam format. If this parameter is provided, the pipeline will start from the peak calling step

II. Command

Note: Please choose the right mode before writing commands.

Mode I: Run the pipeline on your data to get the peak sequences:

Option 1: Start with raw sequencing result (.fastq)

```
$./main --left <read_1_fastq> --right <read_2_fastq> \
[-fa <Human_reference_genome.fa>|-fai <Human_reference_genome.idx>]
```

Option 2: Start with alignment result (.bam)

WARNING: Input as .bam files will work, but will generate several errors that may be misleading to the user. This part of the pipeline is **still in development**. If inputting .bam files please ignore any error messages and proceed as normal.

```
$./main --alignment <alignment.bam>
```

Mode II: Run the pipeline on your data to get the peak sequences and query the database:

Option 1: Start with raw sequencing result (.fastq)

```
$./main --left <read_1_fastq> --right <read_2_fastq> \
[-fa <Human_reference_genome.fa>|-fai <Human_reference_genome.idx>] \
--query -eval 0.01
```

Option 2: Start with alignment result (.bam)

```
$./main --alignment <alignment.bam> --query -eval 0.01
```

Mode III: Only query:

We also have an additional query mode for our pipeline. If a user does not want to install the whole package (i.e. he/she has his own RBP fasta sequences that have been processed) , he/she can simply ask for a query. The syntax for query is:

```
$./query fastafile eval outfile
fasta file: your input fasta file
```

outfile: name of your output file

III. Parameters

Required:

-l | --left	Left-end raw read file in fastq format
-r | --right	Right-end raw read file in fastq format

Optional:

-fa| -fai	Human reference genome/transcriptome in fasta format/ A folder containing the index of the reference
--gtf	Human genome annotation
-v | --version	Software Version
-h | --help	Help Menu
--alignment	Alignment file in bam format
--query	Use if you want to query the database with the output peaks, followed by e-val threshold
-eval	Specify an E-value for BLAST

IV. Example Outputs:

Mode I: Run the pipeline on your data to get the peak sequences:

Peaks.fa: peaks sequences in fasta format

Mode II: Run the pipeline on your data to get the peak sequences and query the database:

Peaks.fa: peak sequences in fasta format
BLASTresult.txt : the blast search results for the peak sequences

Mode III: Only query:

BLASTresult.txt : the blast search results for the peak sequences