Tutorial of aavMPRA

@mengm5@github.com

1 Graphic abstract

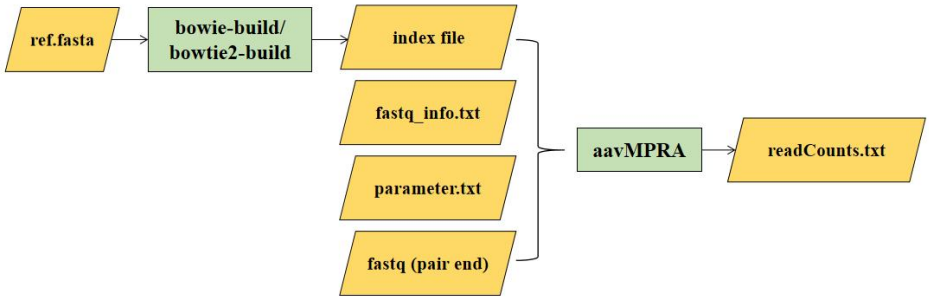

Fig1. Overview of aavMPRA

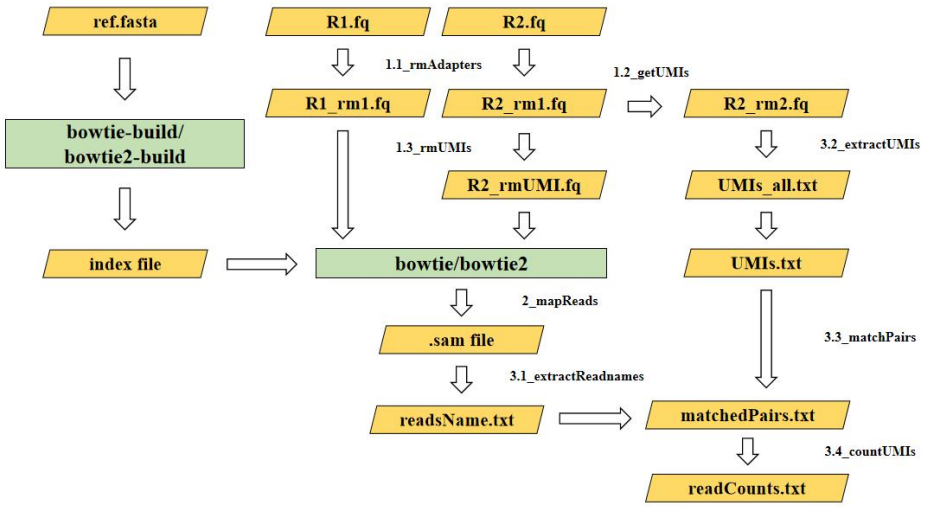

Fig2. Workflow of aavMPRA

2 Reference sequence

2.1 Overall design of library:

P5 + adapter1 + enhancer + adapter2 + UMI + adapter3 + P7[index]P7

AATGATA...CGATCTGCAGATGGCTCTTTGTCCTACACCCTCGGG...CCACAGCCCGGAGCGT

CTTCTGACCGTCGCTTTTTTTTTTTGGGCACAGGAAAGATACTTAGATCG.....AGTCACTGA

CCAATCTCG...TGCTTG

2.2 Structure of read1(R1) and read2(R2):

R1 structure: adapter1 + enhancer_part

CAGATGGCTCTTTGTCCTACACCCTCGGG...

R2 structure: adapter3(reversed) + UMI(reversed) + adapter2(reversed) + enhancer_part(reversed)

AAGTATCTTTCCTGTGCCCAAAAAAAAAAAGCGACGGTCAGAAGACGCTCCGGGCTGTG...

2.3 Steps to correct reads:

1.1_rmAdapters: remove adapter1 and adapter3(reversed)

R1 structure: enhancer_part

R2 structure: UMI(reversed) + adapter2(reversed) + enhancer_part(reversed)

1.2_getUMIs: extract UMI

R1 structure: enhancer_part

R2 structure: UMI(reversed)

1.3_rmUMIs: remove UMI

R2 structure: adapter2(reversed) + enhancer_part(reversed)

2.4 In the step 1.3_rmUMIs, to minimize the loss of sequence information during searching adapters by cutadapt, the adapter2 in R2 will not be trimmed. Hence, the reference sequence for building bowtie/bowtie2 index should contain sequence of adapters for mapping reads. The structure of reference sequence is:

adapter1 + enhancer + adapter2

To build bowtie/bowtie2 index, you should generate .fasta file consist of reference sequences.

3 Data processing

Spending time depending on read depth.

3.1 Ensure that your server runs Linux, we test it on CentOS Linux 7.

3.2 Install conda (https://docs.conda.io/en/latest/miniconda.html) if it is not on your server.

3.3 Clone the repository by the following command:

```
git clone https://github.com/mengm5/aavMPRA.git
```

3.4 Once the repository is cloned, enter the aavMPRA directory and create aavMPRA environment by the following command:

```
cd aavMPRA
conda env create -n aavMPRA -f aavMPRA_environment.yml
```

3.5 Export the path of aavMPRA program by the following command:

```
export PATH=/your_path/aavMPRA/aavMPRA:$PATH
```

3.6 Build the bowtie/bowtie2 index by the following command if it is not been built:

```
bowtie-build ref.fa /your_path/aavMPRA/index/mutagenesis_index/mutagenesis
bowtie2-build ref.fa /your_path/aavMPRA/index/common_index/common
```

The pre-build index is at /your_path/aavMPRA/index for testing.

3.7 Create a fastq_info.txt file. This file contains 4 columns:

1) the first column is paths where fastqs exist,
2) the second is the original names of fastqs,
3) the third is names that you want to change,
4) the forth is used to distinguish read1 (R1) and read2 (R2).

The example file is at /your_path/aavMPRA/data.

Path	Fastq	Rename	Read
/aavMPRA/data	test1_1.fastq.gz	Test1	R1
/aavMPRA/data	test1_2.fastq.gz	Test1	R2
/aavMPRA/data	test2_1.fastq.gz	Test2	R1
/aavMPRA/data	test2_2.fastq.gz	Test2	R2

3.8 Create a parameter.txt file. This file contains 6 columns:

1) the first column is names of steps,

2) the second is names of parameters,

3) the third is the assignment of parameters,

4) the forth is the description of parameters,

5) the fifth is the tools to which the parameters belong,

6) the sixth is the execution mode, the mutagenesis mode uses bowtie to map reads and the common mode uses bowtie2.

The common mode uses bowtie2 because the length of the common library sequence is 400bp, and the length of read1 and read2 can not completely cover the entire sequence. Therefore the bowtie2 can provide better mapping results when mapping longer reads. Meanwhile the mutagenesis mode uses bowtie due to the shorter sequence length of mutagenesis library (230bp).

The example file is at /your_path/aavMPRA/data. You can change the parameters for your requirement.

Steps Name Parameter Explanation Tool Mode

3.9 Once 1) the bowtie/bowtie2 index is built, 2) the fastq_info.txt and parameter.txt are created, run the pipeline by the following command:

```
aavMPRA -o /output_path \ # absolute path is recommended
-f /path_to_fastq_info/fastq_info.txt \
-p /path_to_parameter/parameter.txt \
-m [mutagenesis/common] \ # mutagenesis or common
-i /path_to_index/index_name \
[--gz] # if the fastq file saved as .gz file, please add this parameter.
```

3.10 The aavMPRA will generate .sh file sequentially to execute analysis:

1) 0_softlink.sh: read the fastq_info.txt and generate a file path that contains softlinks of fastqs with standard names.

2) 1.1_rmAdapters.sh, 1.2_getUMIs.sh, 1.3_rmUMIs.sh: correct reads structure by cutadapt.

3) 2_mapReads.sh: map reads to the bowtie/bowtie2 index.

4) 3.1_extractReadnames.sh, 3.2_extractUMIs.sh: extract readnames from .sam

file and UMIs from fastqs.

5) 3.3_matchPairs.sh: combine readnames and UMIs.

6) 3.4_countUMIs.sh, 3.5_mergeSamples.sh: generate readcounts file and merge all readcounts of samples into one.

	input	output
1.1_rmAdapters.sh	xxx_R1.fastq xxx_R2.fastq	xxx_R1_rm1.fastq xxx_R2_rm1.fastq xxx_untrimed_1_rm1.fastq xxx_untrimed_2_rm1.fastq xxx_rmAdapters.log
1.2_getUMIs.sh	xxx_R1_rm1.fastq xxx_R2_rm1.fastq	xxx_R1_rm2.fastq xxx_R2_rm2.fastq xxx_untrimed_1_rm2.fastq xxx_untrimed_2_rm2.fastq xxx_getUMIs.log
1.3_rmUMIs.sh	xxx_R2_rm1.fastq	xxx_R2_rmUMI.fastq xxx_rmUMIs.log
2_mapReads.sh	xxx_R1_rm1.fastq xxx_R2_rmUMI.fastq	xxx.sam xxx_mapReads.log
3.1_extractReadnames.sh	xxx.sam	xxx_readsName.txt
3.2_extractUMI.sh	xxx_R2_rm2.fastq	xxx_UMIs_all.txt
		xxx_UMIs_rm.txt
		xxx_UMIs.txt
3.3_matchPairs.sh	xxx_readsName.txt xxx_UMIs.txt	xxx_matchedPairs.txt
3.4_countUMIs.sh	xxx__matchedPairs.txt	xxx_readCounts.txt
3.5_mergeSamples.sh	xxx,yyy_readCounts.txt	mergedSamplesCount.txt

*xxx_UMIs_all.txt contains all length of short sequences cut from xxx_R2_rm2.fastq, because the length of UMI in our design is 10bp. Hence we remove the ‘not-10bp-UMIs’ and save them into xxx_UMIs_rm.txt while we save the ‘10bp-UMIs’ into xxx_UMIs.txt for next step.

3.11 The aavMPRA will generate a list of directories of result:

1) 0_softlinks: softlinks of all samples (renamed).
2) 1_correctReads: fastqs with corrected reads of all samples.
3) 2_mapReads: sam files with mapped reads of all samples.
4) 3_readCounts: unique UMIs, readCounts files.