Single-Cell Chromatine-ImmunoPrecipitation Data Treatment Pipeline v3.05

Céline Vallot Lab - DEPIC Team
pacome.prompsy@curie.fr

March 12, 2019

Contents

Introduction

This document is a detailed explanation of the single-cell ChIP-seq (scChIPseq) data treatment pipeline developed by Céline Vallot Lab at the Curie Institute to tackle the challenges of this technique.

Briefly, scChIPseq is based on microfluidic techniques allowing the fusion of single cells with droplets containing a single bead. Each bead has a large amount of DNA strands containing a unique barcode composed of three indexes. The DNA fragments of one cell, after ligation on the DNA strands present on the surface of the bead will thus contain a common barcode. After a step of immuno-precipitation on the histone mark of interest, the barcoded DNA fragment are linearly amplified by in-vitro transcription, reversed transcribed using random primers and then once again amplified by PCR to increase the detection. To see more detailed information, go to : . This process is necessary to insert the read 1 primer as well as to increase the detection limits of the original cDNA strands.

The pipeline is taking as input raw paired-end reads, in which the read 1 is containing cDNA and the read 2 (reverse) is containing both the barcode and a small portion of cDNA as described in figure 2. The pipeline main output is one or more count matrices in which the columns are barcodes (equivalent to cells) and the rows are regions defined by the user. Each cell in the matrix indicates the number of reads from a given barcode that falls in the region given by the row. The

pipeline also output a cumulative bigwig file representing the sum of the coverage across all cells, as well as metadata under the form of a HTML report produced through MultiQC.

1 Getting started

1.1 Obtain the pipeline

The scChIPseq pipeline source code can be downloaded from devel for the development branch or master for the master branch. Once you downloaded the source code, the executable is located under $DIR/bin/schip_processing.sh.

1.2 Requirements

The pipeline is coded using Bash and Python 2.7 as well as Python 3.5. The requirements to run the pipeline are :

- Linux/Unix-like machine
- samtools >= 1.9
- fastx-toolkit >= 0.0.14
- bowtie2 >= 2.2.6
- STAR >= 2.6.0
- Python v2.7 :
  - pysam >= 0.11.2.2
  - itertools >= 2.4
  - argparse >= 1.1
  - collections >= 2.4
  - numpy >= 1.13.1
  - scipy >= 0.19.1
  - bx-python >= 0.7.3
  - deeptools >= 3.1.0
- Python v3.5
  - multiqc >= 1.8

1.3 Command line

The command line to run the pipeline is the following (once you cd in the downloaded directory), all of the options are necessary:

```
./bin/schip_processing.sh -f FORWARD.fastq(.gz) -r REVERSE.fastq(.gz)
-c CONFIG -o  outputDirectory
```

The forward and reverse are fastq or compressed fastq raw reads coming from the sequencing platform. The forward reads must contain only genomic DNA while the reverse reads must contain the barcodes. A config file is present at the root of the directory and contains all the option required by the pipeline to run. If needed, the user can change the options by modifying this file. The options are later described in **??**, this documentation is describing the default parameters - running on experimental Curie beads.

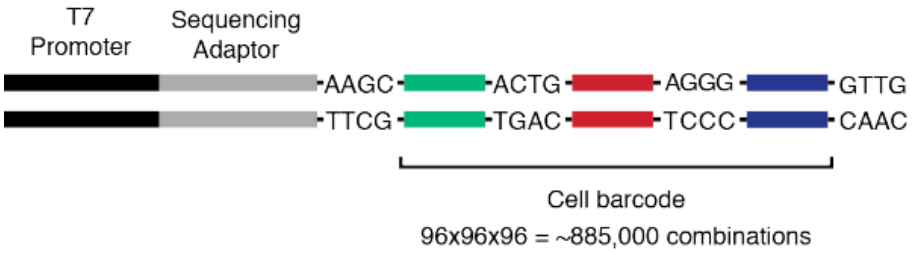

Figure 1: Barcode composition

2 Pipeline, step-by-step

2.1 Align R2 reads on barcode indexes using Bowtie2

In this step, the barcode is retrieved from the reverse reads (R2). As described in figure 2, each barcode is composed of a combination of three indexes, each coming from a 96-index pool. Each index is 16-bp long and the three indexes are linked by two constant linkers, the barcode is therefore 56-bp long. In order to retrieve efficiently the barcodes, taking into account the possible shifts in position or mismatches, the method used is to cut the reverse reads into the three indexes:

- Index 1 : read reverse base 1 to 16
- Index 2 : read reverse base 21 to 36
- Index 3 : read reverse base 41 to 56

The bowtie2 indexes are built from the 3 known pools of indexes, adding the linkers respectively left and right of each index allowing to increase the size of the reference (24-bp) compared to the indexes coming from the reads (16-bp). This is done as follow:

- Reference Index 1 : TTCG-indexes1-TGAC
- Reference Index 2 : TGAC-indexes2-ACCA
- Reference Index 3 : ACCA-indexes3-CAAC

Each index is mapped to it's respective reference, allowing up to 2 mismatches or indels, with bowtie2 using :

```
bowtie2 -x ref_index_X -f indexes_X -L 8 -N 1 --rdg 0,7 --rfg 0,7 --mp 7,7
--ignore-quals --score-min L,0,-1 -t --no-unal --no-hd
```

The options are :

- -L 8 : seed Length fixed to 8, allowing bowtie to construct 2 seeds from the 16-bp
- -N 1 : allowing 1 mismatch per seed
- –score-min L,0,-1 : fixing minimum score to $-1 * index.length = -16$
- –rdg 0,7 –rfg 0,7 –mp 7,7 : setting cost of indels and mismatches to 7
- –ignore-qual : ignoring quality to precisely control the scoring

Since each pool of indexes has been built to maximise the distance between each of the index (it takes 4 substitutions for any index to be mistaken by another index coming from the same pool), we assume that by allowing no more than 2 mismatches/indels, the indexes are not likely to be mistaken by another one, therefore no multimappers are found.

Once each index are successfully mapped, the indexes matching all three indexes are combined to form the barcode. The barcode name is formed by the indexes number mapped in each pool, *e.g. BC010104 means that the index 1 mapped pool1-index1, the index 2 mapped pool2-index1 and the index3 mapped pool3-index4.*

2.2 Trimming R2 reads for genome alignment

The reverse reads (R2) are composed of the barcode in the beginning of the read and the 22 last bases are genomic DNA. Taking this genomic DNA into the alignement allows to align in paired-end mode, which is better at identifying multimappers than single-end mode. Moreover, the mapping of read 2 is very important for unambiguous identification of RT duplicates, as described in section **??**. Fastx trimmer is used to retrieve the last 22bp of R2 in preparation for genome alignment.

2.3 Align paired-end reads on the genome

The pipeline uses STAR 2.6.0 short RNA aligner to map the reads onto the genome. The forward reads are 101bp long and the reverse reads are 22bp long. The pipeline requires a genome index to be built using default parameter on the genome of interest. STAR aligner is ran using the following command :

```
STAR --runMode alignReads --readFilesIn FORWARD.fastq(.gz)
REVERSE_trimmed.fastq(.gz) --genomeDir pathToSTARindex
--alignEndsType EndToEnd --peOverlapNbasesMin 10 --alignIntronMax 1
--alignMatesGapMax 450 --limitGenomeGenerateRAM 25000000000
--outSAMunmapped Within
```

The options are :

- –alignEndsType EndToEnd : align from end to end, not allowing soft-clipping
- –peOverlapNbasesMin 10 : if both reads are overlapping by 10 reads or more, merge them together and retries to map them
- –alignIntronMax 1 : not allowing any introns
- –alignMatesGapMax 450 : allowing a maximum size between the furthest positions of R1 and R2 of 450 (corresponding to tri-nucleosomes)
- –limitGenomeGenerateRAM 25000000000 : limit RAM consumption to 25Go

2.4 Tag reads with barcode

To simplify the downstream process and reduce size of the data, the reads that uniquely map on both end onto the genome are merged into one unique read (read 1). The only information kept from read 2 are the position and sequence, added as tags "XS" and "XD" respectively. Now that the barcode have been identified and the reads have been merged, the barcode information is added to the mapped reads. This is done by joining the alignment file of aligned reads with the assigned barcode file. The barcode name is added in the alignment file as tag "XB" and the barcode sequence is added as a tag "XS".

2.5 Remove PCR duplicates

The particularity of PCR amplification is to create multiple exact copies of the original DNA, 2 copies will therefore map at the exact same location on both ends. The alignment file is sorted by barcode tag, chromosome, R2 position, R1 position respectively. The file is processed line by line (using awk), and PCR duplicates are identified as reads having exactly the same barcode, position R1 and position R2 than the previous read in the file. PCR duplicates identified in this manner are discarded. A ".count" count file is produced containing the number of reads before and after PCR removal.

2.6 Remove RT duplicates

The particularity of RT on the contrary of PCR amplification is that the priming uses random hexamer to introduce read 1 primer sequence. This results in multiple RT copies having different read 1 sequenced but a similar read 2 sequence. In the alignment file this is transcribed by a similar barcode, a similar position R2 but a different position R1. The sorted alignment file, with PCR duplicates removed, is processed stating that a RT duplicate is a read having exactly the

same barcode and position R2 than the previous read, regardless of the R1 position. RT duplicates identified in this manner are discarded. A ".count" count file is produced containing the number of reads after RT removal.

2.7 Generating a bigwig file

This step uses the command "bamCoverage" from DeepTools to create a cumulative coverage .bigwig file from the processed alignment file, using the RPKM normalisation. The BigWig file can easily be visualised into visualisation tools such as IGV to control the overall coverage of the genome and average size of the regions covered.

2.8 Generating the count matrix

This step uses a built-in Python script "sc2count.py" to process the alignment file into a count matrix. The scripts takes as input and options:

- -i alignmentFile.bam : alignment file in BAM format.
- -o file.bw : output file name.
- -s <int> : Number of barcodes in the BAM file. Default: Extracted from the BAM 'CO' field
- -f 500 : minimum reads required to keep a barcode.
- -b 50000 : size of the genomic bin.
- -B file.bed : bed file containing the genomic regions.
- -w False : Use the whole read in the count instead of the 5' end.
- -r False : Do not export bins/features with only zeros in the count table.
- -t "XB" : tag name for the barcode.

As described above, the user has to specify either the size of genome bins to be taken as rows or a path to a personalised bed file to be taken as rows depending on the histone mark of interest. Several binning length and/or bed files can be added in the config file to create several count matrix. The barcode number is automatically calculated by the pipeline. The minimum reads for a barcode to be taken in account is set to 500. This allows to remove barcodes coming from beads with no cells which have ligated to unwanted contaminant DNA.

2.9 Generating multiQC report from the log files

This step uses the log files and a specific multiQC module called scChIPseq to create a metadata report under the HTML format.

3 Setting parameters in the CONFIG file

This section details all the parameters used by the pipeline present in the CONFIG file:

Tools

BOWTIE2_PATH Path to Bowtie2 executable

SAMTOOLS_PATH Path to Samtools executable

PYTHON_PATH Path to Python 2.7

FASTX_PATH Path to Fastx-toolkit executable

STAR_PATH Path to STAR executable

Parameters

NB_PROC Number of processors to use - default = 8

BARCODE_LENGTH Length of the barcode sequence - default = 56

BARCODE_LINKER_LENGTH Length of the barcode plus linker sequence - default = 80

MIN_COUNT_PER_BARCODE_AFTER_RMDUP Minimum reads supporting a barcode to keep it in the final matrix- default = 500

BIN_SIZE Length of the bins used in the count matrix - default = 50000

BED_FEATURES Bed file to use as bins in the count matrix - default = ""

BARCODE_MAPPING_OPTS Bowtie2 options used for the barcode mapping - default = -N 1 -L 8 –rdg 0,7 –rfg 0,7 –mp 7,7 –ignore-quals –score-min L,0,-1 -t –no-unal –no-hd

BARCODE_BOWTIE_IDX_PATH Path to the Bowtie2 indexes - default = /data/users/pprompsy/Annotation/bowtie_2_long/ref_index_

GENOME_MAPPING_OPTS_STAR Mapping options for STAR - default = –alignEndsType EndToEnd –peOverlapNbasesMin 10 –alignIntronMax 1 –alignMatesGapMax 450 –limitGenomeGenerateRAM 25000000000 –outSAMunmapped Within

GENOME_IDX_PATH_STAR Path to the STAR index directory - default = /data/annotations/pipelines/Human/hg38/indexes/STAR

TMP_DIR Path to the tmp directory - default = /data/tmp/pprompsy/tmp/

4 Appendix

4.1 Single-cell ChIP-seq procedure: from the single cell to the count matrix

X. Single-cell ChIP-seq procedure: from the single cell to the count matrix

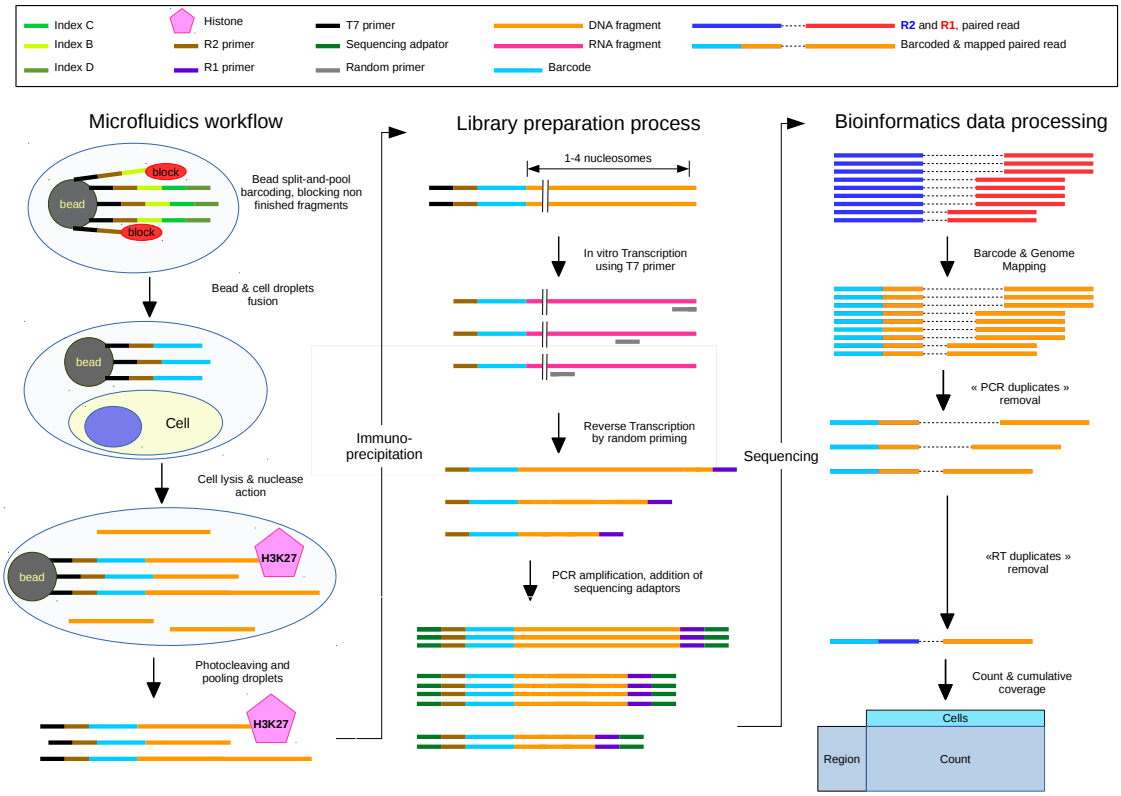

Figure 2: Description of the experimental and in-silico pipeline