ChimPipe Documentation

Release 0.9.4

Bernardo Rodríguez-Martín and Sarah Djebali

January 05, 2017

Contents

ChimPipe is a computational method for the detection of novel transcription-induced chimeric transcripts and fusion genes from Illumina Paired-End RNA-seq data. It combines junction spanning and paired-end read information to accurately detect chimeric splice junctions at base-pair resolution.

ChimPipe source code is freely available in our GitHub repository.

CHAPTER 1

Table of contents:

1.1 Installation

1.1.1 Download

You can do it from our GitHub repository in two different ways:

1. Go to the releases tab and download the latest release.
2. Clone the git repository in case you want the latest version of the code:

```
# Move into the folder in which you want to clone the repositoy.
$ cd ~/apps
# Clone it.
$ git clone https://github.com/Chimera-tools/ChimPipe.git
```

ChimPipe does not require any further installation step. It already comes with precompiled gemtools binaries. It is written in Bash and Awk and can be run as a standalone application on diverse Linux systems.

1.1.2 Requirements

Hardware:

- 64-bits CPU.
- RAM: ~40G for 100 million illumina paired-end reads.

Software:

- 64-bit Linux System.
- Bedtools v2.20.1 or higher.
- Samtools v0.1.19 or higher.
- Blast v2.2.29+ or higher (only if you want to generate your own similarity text files, see **Gene pair similarity file** in the *Manual* section).

1.2 Manual

This section describes ChimPipe input files, how to run ChimPipe and interpret the output files.

1.2.1 Input

ChimPipe takes as input 3-4 mandatory files and 1 optional file.

Mandatory:

- *Paired-end RNA-seq reads*
- *Genome index*
- *Reference annotation*
- *Transcriptome index* (only in FASTQ as input mode)

Optional:

- *Gene pair similarity*

Paired-end RNA-seq reads (FASTQ or BAM)

ChimPipe makes use of Illumina paired-end RNA sequencing data to identify chimeric transcripts. It can deal with both unaligned and pre-aligned reads in the standard formats:

A) FASTQ:

- Two independent FASTQ files are required for mate 1 and 2 reads, respectively.
- The read identifiers should follow any of the 2 Illumina conventions to specify which member of the pair the reads are. Check our *FAQ* section for more information.

B) BAM:

- In BAM as input mode the first mapping step is skipped.
- A single BAM file with pre-aligned reads has to be given as input.
- The BAM file should contain unmapped reads. ChimPipe will attempt to split-map them across different chromosomes and strands to identify chimeric splice junctions in a second mapping step.
- The **NH** (Number of hits) and **NM** (Edit distance) fields are required. Most popular mappers provide this information. Check SAM specifications for more information.

For more information about the supported input RNA-seq data check our *FAQ* section.

Note: ChimPipe results are highly dependent on the mapping step. BAM files should have been generated with a mapper able to align read-pairs across different chromosomes and strands. If you are unsure about that, please convert your BAM file to FASTQ and run ChimPipe in FASTQ as input mode.

Genome index

An indexed reference genome in GEM format is required for first and second mapping steps.

We supply **pre-generated indices** for Human, Mouse and Drosophila in the *Downloads* section.

If you aim to execute ChimPipe with your own genome you just need to use the *GEMtools indexer* (at `ChimPipe/bin/gemtools-1.7.1-i3/gemtools`):

```
$ gemtools index -i genome.fa
```

It will produce 3 files in the directory where your genome is located:

- **genome.gem** – indexed genome in GEM format (mandatory).
- genome.hash – genome hash table.
- genome.log – log file.

Tip: If your machine has more than one CPU, it is recommended to run the indexer with multiple threads. Use the option `-t <threads>`, where **threads** is the number of available CPUs.

Reference annotation

ChimPipe requires a reference gene annotation in the standard GTF format:

- The annotation has to contain all the annotated exons for a given species.
- Any other feature is accepted, Introns, UTR.., but they will not be considered for chimera detection.
- The gtf file has to be sorted by chr, then start and end.
- The following mandatory tag,value pairs are required in the attribute field: gene_id "X"; transcript_id "Y";
- Two optional tag,value pairs will be considered if provided: "gene_name" and "gene_type". There can be additional tag,value pairs, but they will not be taken into account.
- The order of the pairs does not matter, but the tag ids have to be exactly the ones as described since they are used to parse the gtf and extract the information.

E.g:

```
# Example of an annotated human exon in GTF format.
# The attributes are gene_id (mandatory), gene type and gene name (optional)

chr1    HAVANA  exon    69091   70008   .       +       .
gene_id "ENSG00000186092.4"; gene_type "protein_coding"; gene_name "OR4F5";
```

Tip: We have extensively tested and applied ChimPipe to analyse human and mouse RNA-seq data with Gencode annotations. So, we suggest to use Gencode if possible.

Transcriptome index (only in FASTQ as input mode)

A transcriptome index containing all the annotated exon-exon junctions for each transcript is needed for the first mapping step.

We supply **pre-generated transcriptome indices** for Human, Mouse and Drosophila annotations in the *Downloads* section.

If you aim to execute ChimPipe with your own annotation you just need to use the *GEMtools transcriptome indexer* (at `ChimPipe/bin/gemtools-1.7.1-i3/gemtools`) on your previously generated GEM indexed genome and annotation, as indicated below:

```
$ gemtools t-index -i genome.gem -a annotation.gtf
```

It will produce 5 files in your current working directory:

- annotation.gtf.junctions – annotated splice junctions coordinates.
- annotation.gtf.junctions.fa – annotated splice junctions sequences.
- **annotation.gtf.junctions.gem** – transcriptome index in GEM format.
- **annotation.gtf.junctions.keys** – keys to convert from transcriptome to genome coordinates.
- annotation.gtf.junctions.log – log file.

Tip: If your machine has more than one CPU it is recommended to run the indexer with multiple threads. Use the option `-t <threads>`, where **threads** is the number of available CPUs.

Gene pair similarity file (Optional)

ChimPipe filters out artefactual chimeric junctions involving genes with high exonic sequence homology. These genes are prone to produce spurious misaligned split-reads connecting them.

Before applying this filter, ChimPipe has to compute a similarity matrix between every annotated gene pair.

This step takes around 45'-60' depending on the annotation. So, in case you plan to run many samples with the same annotation, it is recommended to choose one of these two options:

A) Execute ChimPipe with a single sample and then reuse the generated matrix (`${outDir}/GnSimilarity/${annotation_id}.similarity.txt`) to run the other samples

B) Pre-compute the matrix executing `ChimPipe/src/bash/similarity_bt_gnpairs.sh` as follows and run all your samples:

```
$ bash similarity_bt_gnpairs.sh annot.gtf genome.gem
```

Either if you use A) or B) you can provide the matrix to ChimPipe with the option `--similarity-gene-pairs <MATRIX TEXT FILE>`. Otherwise, ChimPipe will generate the same matrix per each sample.

Note, we supply **pre-generated matrices** for Human, Mouse and Drosophila in the *Downloads* section.

Warning: Make sure you run ChimPipe with a similarity matrix generated from the same reference annotation and genome you are using.

1.2.2 Execute ChimPipe

1. Setting up the environment

As explained in the *Installation* section, to execute ChimPipe you need to have BEDtools, SAMtools and Blast binaries installed in your linux environment.

```
$ # Example about how to export your binaries under your environmnet
$ export PATH=<BEDTOOLS_PATH>:<SAMTOOLS_PATH><BLAST_PATH>:$PATH
$ export PATH=~/bin/bedtools2-2.20.1/bin:~/bin/samtools-0.1.19:~/bin/blastn:$PATH
```

2. Running ChimPipe

ChimPipe has two different running modes:

A) FASTQ

```
ChimPipe.sh --fastq_1 <mate1_fastq> --fastq_2 <mate2_fastq> -g <genome_index>
-a <annotation> -t <transcriptome_index> -k <transcriptome_keys> [OPTIONS]
```

All these files and parameters given as input to ChimPipe are **mandatory arguments**. See bellow a detailed descripion:

```
--fastq_1                       <FASTQ>         First mate sequencing reads.
                                                It can be gzip compressed [.gz].
--fastq_2                       <FASTQ>         Second mate sequencing reads.
                                                It can be gzip compressed [.gz].
-g|--genome-index               <GEM>           Reference genome index.
-a|--annotation                 <GTF>           Reference gene annotation.
-t|--transcriptome-index        <GEM>           Annotated transcriptome index.
-k|--transcriptome-keys         <KEYS>          Transcriptome to genome index
                                                conversion keys.
--sample-id                     <STRING>        Sample identifier.
```

B) BAM

```
ChimPipe.sh --bam <bam> -g <genome_index> -a <annotation> [OPTIONS]
```

```
--bam                           <BAM>           Pre-aligned reads in BAM format.
-g|--genome-index               <GEM>           Reference genome index.
-a|--annotation                 <GTF>           Reference gene annotation.
--sample-id                     <STRING>        Sample identifier.
```

Optional arguments. Do `ChimPipe.sh -h or --help` to see a short help with the main options. You can also do `ChimPipe.sh --full-help` to see the all the possible options.

Tip: If your machine has more than one CPU it is recommended to **run ChimPipe with multiple threads** (at least 4). It will speed up the mapping steps. Use the option `-t|--threads <threads>`, where threads is the number of CPUs available.

Note: **Checkpoints**: ChimPipe has a checkpoint in every step. So, in case the program fail at one step. You can restart the analysis from this step. You just need to make sure that you removed the files generated in the step where the pipeline failed.

1.2.3 Output

By default, ChimPipe produces 4 main output files:

- First mapping BAM.
- Second mapping MAP.
- Final chimeric junctions.
- Discarded chimeric junctions.

Tip: If you want to keep intermediate output files, run ChimPipe with the `--no-cleanup` option.

First mapping BAM file

BAM file (`${outDir}/MappingPhase/FirstMapping/${sample_id}_firstMap.bam`) containing the reads mapped in the genome, transcriptome and *de novo* transcriptome with the GEMtools RNA-seq pipeline.

BAM is the standandard format for aligned RNA-seq reads, meaning that most analysis tools work with this format. The bam file produced can therefore be used to do other downstream analyses such as gene and transcript expression quantification.

Second mapping MAP file

MAP file (`${outDir}/MappingPhase/SecondMapping/${sample_id}_secondMap.map`) containing the reads split-mapped in the genome allowing for interchromosomal, different strand and unexpected genomic order mappings.

Final and filtered chimeric junction files

Two tabular text files with the detected (`${outDir}/chimericJunctions_${sample_id}.txt`) and filtered out (`${outDir}/chimericJunctions_filtered_${sample_id}.txt`) chimeric splice junctions from your RNA-seq dataset. They consist on rows of 35 fields, where each row corresponds to a chimeric junction and each field contains a piece of information about the chimera. Here is a brief description of the 35 fields (most relevant fields highlighted in bold):

1. **juncCoord** - Position of the chimeric splice junction in the genome described as follows: chrA"_"coordA"_"strandA":"chrB"_"coordB"_"strandB. E. g. `chr4_90653092_+:chr17_22023757_-` is a chimeric junction between the position 90653092 of chromosome 4 in plus strand, and the position 22023757 of chromosome chr17 in minus strand. Junction coordinates defined using 1-based system.
2. **type** - Type of chimeric splice junction. Junctions classified in 5 different categories:
    - readthrough: donor and acceptor sites within the same chromosome, strand and within less than 100.000 base pairs.
    - intrachromosomal: donor and acceptor sites within the same chromosome, strand and in separated by more than 100.000 base pairs.
    - inverted: donor site downstream than acceptor site (opposite as expected) and both in the same chromosome.
    - interstrand: donor and acceptor sites within the same chromosome but in different strand.
    - interchromosomal: donor and acceptor sites in different chromosomes.
3. **filtered** - Flag to specify if the chimeric junction has been filtered out (1) or not (0).
4. **reason** - List of filters the chimeric junction failed to pass. There is a tag per filter:
    - totalSupport.
    - spanningReads.
    - consistentPE.
    - percStag.
    - percMulti.
    - percInconsistentPE.
    - similarity.

- biotype.

5. **nbTotal(spanning+consistent)** - Total supporting evidences (spanning reads + consistent paired-ends).
6. **nbSpanningReads** - Number of split-reads spanning the chimeric splice junction.
7. nbStaggered - Number of spanning split-reads aligning at different positions.
8. percStaggered - Percentage of spanning split-reads that aligns at different positions.
9. nbMulti - Number of multimapped split-reads spanning the chimeric junction.
10. percMulti - Percentage of spanning split-reads mapping in multiple locations.
11. **nbConsistentPE** - Number of discordant paired-end consistent with the chimeric junction.
12. nbInconsistentPE - Number of discordant paired-end inconsistent with the chimeric junction.
13. percInconsistentPE - Percentage of discordant paired-end inconsistent with the chimeric junction.
14. overlapA - Percentage of overlap between the 5' split-read cluster and the annotated exon.
15. overlapB - Percentage of overlap between the 3' split-read cluster and the annotated exon
16. **distExonBoundaryA** - Distance between the chimeric junction donor site and the exon boundary (annotated donor).
17. **distExonBoundaryB** - Distance between the chimeric junction acceptor site and the exon boundary (annotated acceptor).
18. blastAlignLen - Maximum length of the BLAST alignment between all the transcripts of the gene pairs connected by the chimeric junction. "na" if no blast hit found.
19. blastAlignSim - Maximum percent of similarity in the BLAST alignment between the transcript with the longest BLAST alignment. "na" if no blast hit found.
20. **donorSS** - Splice donor site sequence.
21. **acceptorSS** - Slice acceptor site sequence.
22. beg - Split-reads cluster 5' coordinates.
23. end - Split-reads cluster 3' coordinates.
24. sameChrStr - Flag to specify if the connected gene pairs are in the same cromosome and strand (1) or not (0).
25. okGxOrder - Flag to specify if the connected gene pairs are in genomic order (1) or not (0). "na" in case the samechrstr field was 0 (being in genomic order means that the donor gene is located in 5' while the acceptor in 3'. This means that if both genes are in the plus strand, the genomic coordinates of the first gene are lower than the ones for the second one. For genes in the minus it is the opposite).
26. dist - Distance between the chimeric junction splice sites. "na" in case the "sameChrStr" field was 0.
27. **gnIdsA** - 5' gene/s involved in the chimeric transcript.
28. **gnIdsB** - 3' gene/s involved in the chimeric transcript.
29. **gnNamesA** - Name of the genes in the field *gnIdsA*. "na" if unknown.
30. **gnNamesB** - Name of the genes in the field *gnIdsB*. "na" if unknown.
31. **gnTypesA** - Biotype of the genes in the field *gnIdsA*. "na" if unknown.
32. **gnTypesB** - Biotype of the genes in the field *gnIdsB*. "na" if unknown.
33. juncSpanningReadsIds - Identifiers of the split-reads spanning the chimeric splice junction.
34. consistentPEIds - Identifiers of the paired-ends consistent with the chimeric splice junction.

35. inconsistentPEIds - Identifiers of the paired-ends inconsistent with the chimeric splice junction.

1.3 Tutorial

The follow tutorial explains how to execute ChimPipe to identify already known fusion genes on the MCF-7 Breast cancer cell line.

A total of 6 different fusion genes have been experimentally validated by Edgren et al. (Genome Biology, 2011) and Kangaspeska et al. (PLOSone, 2012). 5 of them are detected by ChimPipe:

Fusion Gene	Chimeric splice-junctions	Detected
BCAS4-BCAS3	chr20_49411710_+:chr17_59445688_+	Yes
ARFGEF2-SULF2	chr20_47538547_+:chr20_46365686_-	Yes
RPS6KB1-VMP1	chr17_57992064_+:chr17_57917129_+	Yes
GCN1L1-MSI1	chr12_120628101_-:chr12_120785317_-	No
AC099850.1-VMP1	chr17_57184952_+:chr17_57915656_+	Yes
SMARCA4-CARM1	chr19_11097269_+:chr19_11015627_+	Yes

1.3.1 1. Download ChimPipe

Check our *Installation* section for details about how to download and install ChimPipe.

1.3.2 2. Download all-in-one package

This package contains all the needed files for executing ChimPipe on MCF-7 RNA-seq data in the "input" folder:

- Paired-end RNA-seq data: MCF-7_1.fastq.gz and MCF-7_2.fastq.gz
- Gencode 19 annotation: gencode.v19.annotation.long.gtf
- Genome index: Homo_sapiens.GRCh37.chromosomes.chr.M.gem
- Transcriptome index: gencode.v19.annotation.long.gtf.junctions.gem
- Transcriptome keys: gencode.v19.annotation.long.gtf.junctions.keys
- Similarity matrix: gencode.v19.annotation.long.similarity.txt

Click here to download.

1.3.3 3. Execute ChimPipe on MCF-7

```
ChimPipe.sh --fastq_1 MCF-7_1.fastq.gz --fastq_2 MCF-7_2.fastq.gz
-g Homo_sapiens.GRCh37.chromosomes.chr.M.gem -a gencode.v19.annotation.long.gtf
-t gencode.v19.annotation.long.gtf.junctions.gem -k gencode.v19.annotation.long.gtf.junctions.keys
--sample-id MCF-7 --threads 4 --similarity-gene-pairs gencode.v19.annotation.long.similarity.txt
```

1.3.4 4. Examine your output

ChimPipe identifies 13 chimeric transcript, including 5/6 validated cases and 3 readthrough transcripts.

We provide ChimPipe output for MCF-7 on the "output" folder. If everything is ok, your results should be identical as the ones provided.

Check our *Manual* section for an explanation about ChimPipe output files.

1.4 FAQ

1.4.1 Quality offset encoding

ChimPipe accepts FASTQ with both 33 and 64 (old Illumina machines) quality encodings. It will automatically infer the offset from your input reads, so you do not need to specify it as a parameter.

1.4.2 RNA-seq protocols accepted

Different protocols can be used to generate a RNA-seq library. Currently, ChimPipe can handle data generated with the following protocols:

- **Non strand-specific protocols**. Information about from which strand the transcript is transcribed is not available.
- **Strand-specific protocols**:
  - **MATE1_SENSE**. Mate 1 directly sequenced from the transcript sequence and mate 2 from the reverse complement of the transcript sequence.
  - **MATE2_SENSE**. Mate 1 sequenced from the reverse complement of the transcript sequence and mate 2 directly from the transcript sequence.

ChimPipe is able to infer the protocol used to produce your data. It takes a subset of 1M of mapped reads and compares the mapping strand with the strand of the annotated gene where they map. Optionally, you can supply this information and skip this step with the option `-l|--seq-library <library>`.

1.4.3 Read identifier format

ChimPipe requires the read identifiers to meet one of the two Illumina standards to specify which member of the pair the read is:

- **CASAVA lower than v1.8**. The identifier has to end in /1 and /2 for mate 1 and mate 2, respectively. E. g.:

```
$ # Mate 1
@SRR018259.1_BI:080831_SL-XAN_0004_30BV1AAXX:5:1:708:1868#0/1
GTAACATATTCACAGACATGTCAGTGTGCACTCAGGAACACTGGTTTCATT
+
IIIIIIIIIIIIIII>=CII=8=H032/-++D+'.@)2+4/+)1'4.#"*.

$ # Mate 2
@SRR018259.1_BI:080831_SL-XAN_0004_30BV1AAXX:5:1:708:1868#0/2
CAGATGGCATTGCAGAACTAAAAAAGCAAGCTGAATCAGTGTTAGATCTCC
+
IIIGIIIIIIIIIFI:>DID<AFH=>78@I41I055549.?+.42-'%**'
```

- **CASAVA v1.8 or higher**. The identifier consists on two strings separated by a space with the second one specifying the mate in the first digit. E.g:

```
# Mate 1
@seq.1_set1:140:D034KACXX:1:2105:18345:62400 1:N:0:
CCCAGCCTGTTTCCCTGCCTGGGAAACTAGAAAGAGGGCTTCTTTCTCCT
+
```

```
IJJJIIIIIGIGIEBHHHGFFFF=CDEEEEEDDDDCCCDDA?55><CBBB

# Mate 2
@seq.1_set1:140:D034KACXX:1:2105:18345:62400 2:N:0:
GCACCCTTCACTCCCTCCCTTGGGCGCCTCCCTCCCGAGGGTAGGGACCC
+
FFHIJJCHIIHBHIIIAHFFFFFCDEDEECDBB;??@CD?CCCCCCC@CC
```

In case your FASTQ files do not meet this format you should modify the identifier. Awk is a perfect tool for such kind of problems. E.g: we downloaded a pair of FASTQ files from the NCBI Sequence Read archive:

```
# Mate 1 reads without a proper identifier.

@SRR018259.1 BI:080831_SL-XAN_0004_30BV1AAXX:5:1:708:1868 length=51
GTAACATATTCACAGACATGTCAGTGTGCACTCAGGAACACTGGTTTCATT
+SRR018259.1 BI:080831_SL-XAN_0004_30BV1AAXX:5:1:708:1868 length=51
IIIIIIIIIIIIIII>=CII=8=H032/-++D+'.@)2+4/+)1'4.#"*.

# No worries, we can use awk to fix the FASTQ file

$ awk '{if (NR%2==1){print "@"$2"#0/1"} else {print $0}}' dataset_1.fastq

@BI:080831_SL-XAN_0004_30BV1AAXX:5:1:708:1868#0/1
GTAACATATTCACAGACATGTCAGTGTGCACTCAGGAACACTGGTTTCATT
@BI:080831_SL-XAN_0004_30BV1AAXX:5:1:708:1868#0/1
IIIIIIIIIIIIIII>=CII=8=H032/-++D+'.@)2+4/+)1'4.#"*.

$ # Finally, Apply the same procedure for mate 2 FASTQ
```

1.5 Downloads

1.5.1 GEM genome indices

Homo sapiens

- GRCh37/hg19 (February 2009)

Mus musculus

- NCBI37/mm9 (July 2007)

Drosophila melanogaster

- FlyBase genome v5.46 (July 2012)

1.5.2 GEM transcriptome indices

H. sapiens

- Gencode v19 (July 2013, hg19) long transcripts.
- Gencode v10 (July 2011, hg19) long transcripts.

Note: The long transcript group excludes ribosomal, micro, transfer, small cytoplasmic, small nucleolar and small nuclear RNAs.

M. musculus

- Ensembl v65 (December 2011, mm9) long transcripts.

D. melanogaster

- FlyBase annotation v5.46 (July 2012, v5.46)

1.5.3 Similarity between gene pair matrices

H. sapiens

- Gencode v19 similarity matrix for long transcripts (GRCh37/hg19).
- Gencode v10 similarity matrix for long transcripts (GRCh37/hg19).

M. musculus

- Ensembl v65 similarity matrix for long transcripts (GRCh37/mm9).

1.6 Contact

Please feel free to join the ChimPipe's mailing list in case you have a question, you want to report an issue or request a feature.

You can also directly contact us at: chimpipe.pipeline@gmail.com

CHAPTER 2

License

ChimPipe is distributed under the GPLv3.