Nicolas Servant 05 March 2014

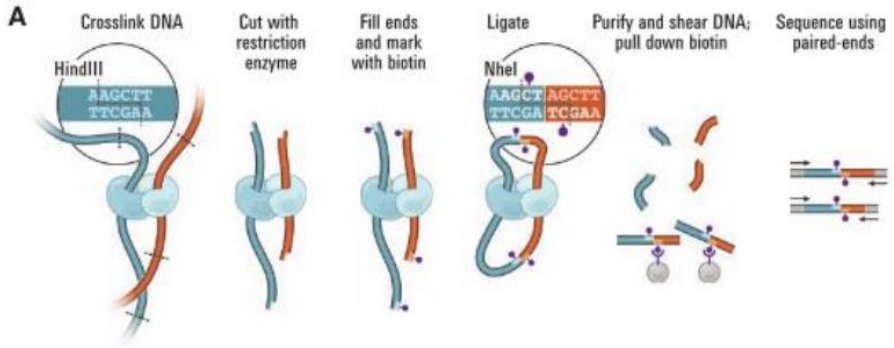
Hi-C pipeline

Goal

The goal of this pipeline is to go from raw sequencing reads to Hi-C contact maps

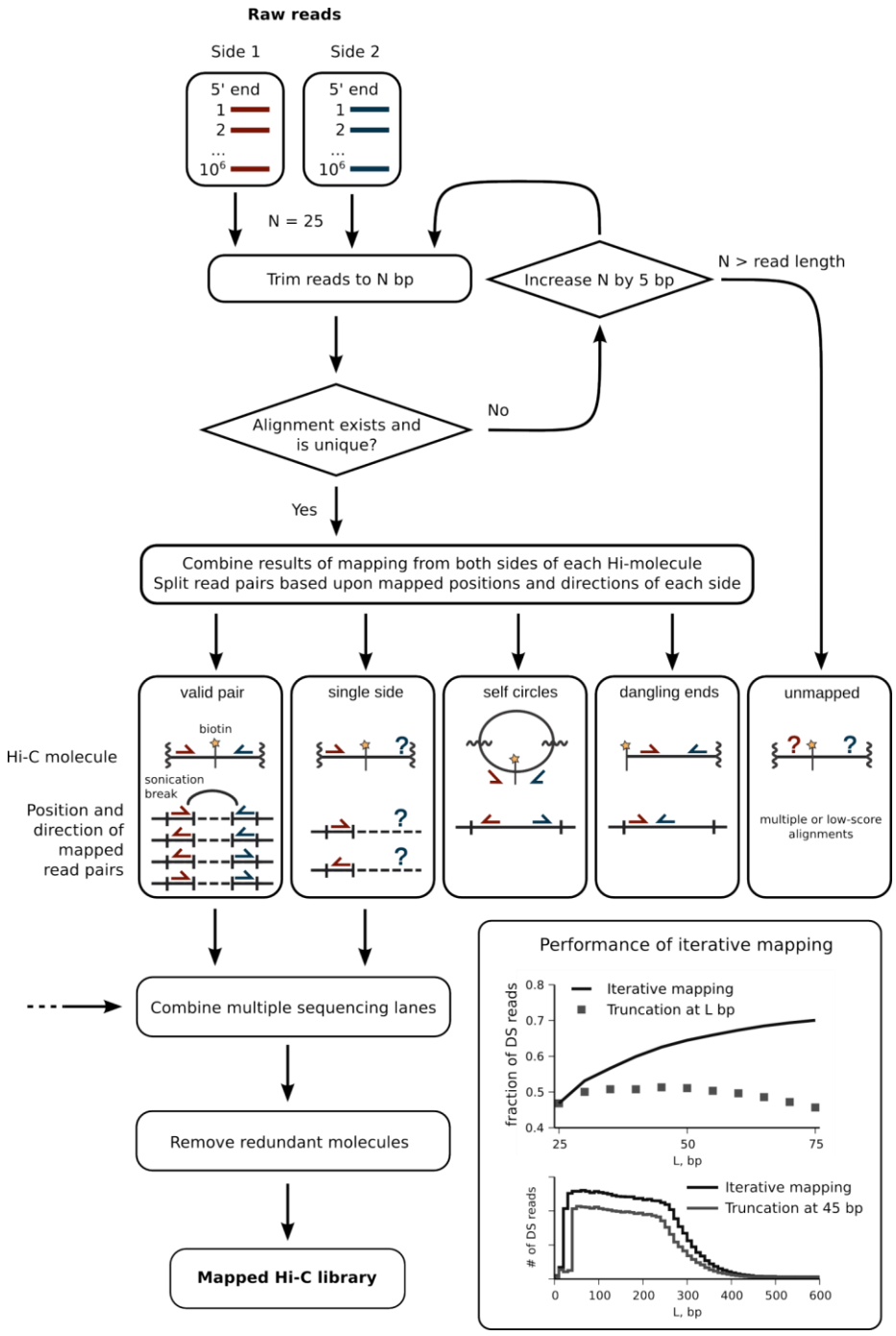
Hi-C in brief

Chromosome folding can bring distant elements, such as promoters and enhancers, close together in space, and thus affecting genome regulation or gene transcription. Recent methods based on high-throughput 3C techniques were recently developed to identify long-range genomic interactions within a genomic region of a few megabases (5C) or at the scale of the entire genome (Hi-C). The Hi-C protocol involves formaldehyde-fixing cells to create DNA-protein bonds that cross-link interacting DNA loci. The fixed DNA is then digested with one restriction enzyme, fragmenting the molecule but maintaining their interaction. Next, a ligation is performed to produce a library of ligation products that were close to each other in the nucleus. The library is then further fragmented, either by using a second restriction enzyme or by sonication, leading to DNA fragments ready to be sequenced (Lieberman-Aiden et al. 2009).



Main steps of the data processing

The Hi-C processing was presented in a recent paper from Imakaev et al. (NatGen. 2012). The different steps of the workflow are presented in the following figure from their paper. It is important to note that their codes are available but not optimized and difficult to reuse (http://mirnylab.bitbucket.org/hiclib/mapping.html). The goal of this pipeline is to process the Hi-C data in a similar way.



1. Raw reads

As illustrated at the end of the protocol from Lieberman-Aiden et al., ligation fragments are sequenced from both ends. Valid Hi-C pairs comprise two DNA fragments from different regions of the genome ligated together. Typically, a forward read (R1) maps to one ligation fragment; the reverse read (R2) maps to the other. The Hi-C contact is represented by the number of reads supporting the same interaction. The number of reads for a single experiment depends on the desired resolution. Typically around 1.10^9 reads are required for a 40-20kb resolution.

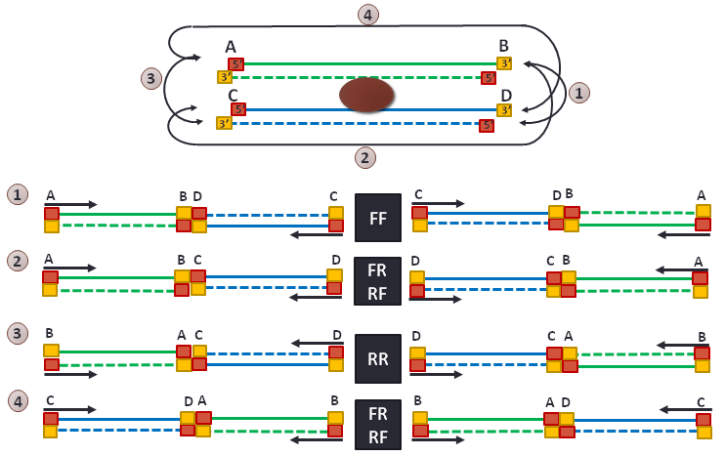
2. Reads mapping

The reads mapping consist at aligning raw reads from the sequencer to a reference genome. The Hi-C protocol is based on paired-end sequencing, meaning that for each DNA fragment, both fragment ends are sequenced. The standard way of aligning such data is to align them together on a reference genome, using both alignments to efficiently choose the reads position on the genome. However, for the 3C products we expect to find contacts between distal elements, and we thus need to separately align both R1 and R2 ends. The main difficulty of this mapping is that Hi-C ligation junction may be found within the sequenced region. Such reads will most likely be removed from the Hi-C pipeline during the mapping process, hereby losing potentially valid data. To address this issue, Imakaev et al. propose an iterative mapping strategy which consists at aligning truncated reads of a given size and to iteratively increase the size of the trimming regions until the reads was mapped. A final step of mapped reads filtering can be achieved to remove multiple reads alignments.

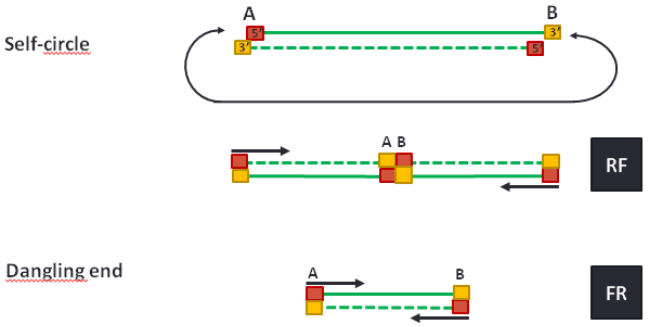
3. Reads assignment to restriction fragments

The majority of aligned paired reads are most likely valid Hi-C products, but a substantial minority are probably not and should be removed. The goal of this step is to reconstruct the ligation products and to filter the different cases of ligation product as illustrated on the figure from the Imakaev paper. This filtering is mainly based on the assignment of reads on its restriction fragment, the proximity with the restriction sites, and the orientation of the paired end sequencing.

- **Interactions**: Valid 3C product. As illustrated in the next figure, different ligation products are possible. The same proportion of each class of interaction is expected.



- **Dangling end:** correspond to fragment which were not circularized. Both reads map the same restriction enzyme.
- **Self-circle:** a DNA fragment cut with the restriction enzyme but circularises, ligating to itself. Both reads map on the same restriction fragment with opposite direction.



- **Single:** Only one read of the pairs is aligned.

- **Error:** All other cases. Mainly fragment ends map to expected locations within the genome, but the orientation of the cut sites do not correspond to any of the aforementioned categories.

4. Generation of contact matrices

At this step, PCR duplicates can be removed from the paired end read merged from multiple input files. Once valid contact pairs are selected and merged per sample, the contact matrix can be generated for a given resolution. For a pairs of genomic window, the interaction is summarized by the number of reads supporting an interaction between fragment included in the given window ranges. The final matrix should be a binned matrix genome wide, or per couple of chromosome. It should be a symmetrical matrix in most of the cases, and sparse for high resolution.

5. Data normalization

Distinct biases have already been described from Hi-C protocol. More details can be found in the Imakaev et al. paper, highlighting the need for dedicated normalization method. Currently the most popular normalization method named ICE, for iterative correction and eigenvector decomposition is applied on the raw contact matrices.

Description of the pipeline v1.0

The current pipeline is available under SVN. A test dataset is available in the branches folder.

https://svn.curie.fr/svn/U900/pf-integration/ngs/tools/HiC-pip

This pipeline is mainly developed in Perl and bash. Some functionalities of the HiTC R package are also used. The main steps of the pipeline can be run through a Makefile, allowing to independently running the different steps of the pipeline. The current version is applied per sample, and the mapping is the only parallelize step as presented in the following general workflow.

1. Running the pipeline

All options are defined in the config.txt file. The user then has to put its input fastq files in the rawdata folder. A sub-folder is required to provide the sample names (i.e. rawdata/test_sample/*.fastq). Multiple fastq files for a given samples are thus possible.

To run the pipeline, use the following command:

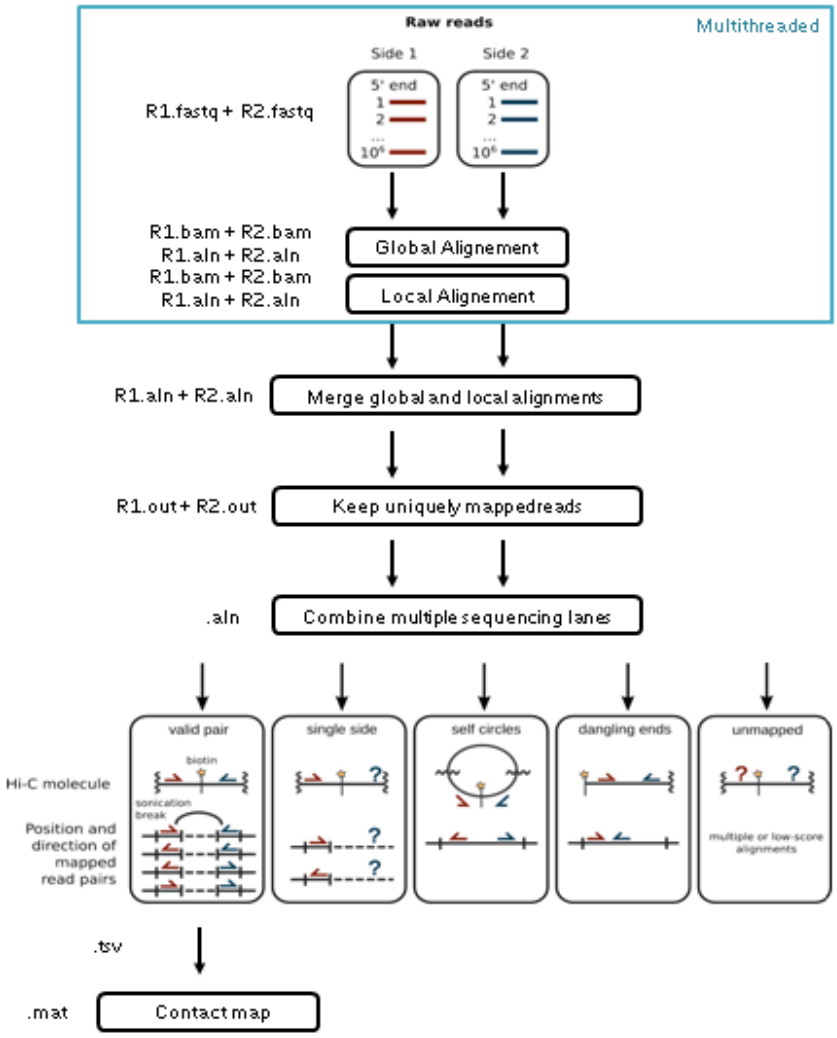
make --file Makefile CONFIG_FILE=config.txt > log.txt

All steps of the pipeline are managed by a Makefile, meaning that the user can easily run a single step of the pipeline. As an example, the mapping can be run using:

make --file Makefile CONFIG_FILE=config.txt mapping > log.txt

2. Description of the workflow

The current workflow of the pipeline can be summarized as follow:



Reads mapping.

In the current version of the pipeline we propose to address the iterative mapping using a local alignment strategy. Thus 3 mapping steps are performed; a first global mapping (both R1 and R2 ends in parallel), a second local mapping of the unmapped reads (both R1 and R2 ends in parallel), and a merge of both alignments. This step is performed using the bowtie2 aligner in global or local mode.

Input: The input files in fastq format from the rawdata folder.

Output: Mapping results (BAM files) are then respectively available in the bowtie_results/bwt2_global, and bwt2_local folders.

Reads filtering and sample merging

The merge aligned results are available in the bwt2 folder using another format (.aln). This format corresponds to the Novoalign format and is mandatory to run the filtering step. Multiple aligned files from the same samples are merged at the step.

Input: BAM (+ aln) results files from the mapping are merged in the bowtie_results/bwt2 folder.

Output: .out files with one read per line. One file per reads tag per sample is generated. Only unique reads are kept. Output is written in the bwt2 folder.

Reads assignment to restriction fragments

This script was developed by Bryan Lajoie in Perl. The reads are assigned to a restriction fragment, and then classify as true interaction, self-circle, dangling end, or error.

Input: The input must fit the Novoalign aligner output (.out files). The list of restriction fragments is also required.

Output: All pairs of restriction fragments are written in a separate file according to their classification.

Generation of contact maps

The final step is to generate the contact matrices from the interaction files. A couple of parameters are available for this step such as the bin size and the bin step.

Input: Interaction data per restriction fragments.

Output: Matrix txt files.

Contact matrix normalization

The normalization is applied with the HiTC R package for each inter-chromosomal contact map.