GITAR: An open source tool for analysis and visualization of Hi-C data

Riccardo Calandrelli[1,*], Qiuyang Wu[2], Jihong Guan[2], and Sheng Zhong[1]

[1] Department of Bioengineering, University of California San Diego, La Jolla, CA 92093, USA.
[2] Department of Computer Science and Technology, Tongji University, Shanghai, China.
* Correspondence: rcalandrelli@eng.ucsd.edu

ABSTRACT

Interactions between chromatin segments play a large role in functional genomic assays and developments in genomic interaction detection methods have shown interacting topological domains within the genome. Among these methods, Hi-C plays a key role. Albeit the presence of several software to process and visualize Hi-C data, a tool to perform a comprehensive analysis including also data pre-processing and topological domains computation was still missing. To address this need we developed GITAR (Genome Interaction Tools and Resources).
GITAR is composed of two modules: 1) HiCtool, a Python library to process and visualize Hi-C data, including topologically associating domains (TADs) analysis and 2) Processed data, a large collection of human and mouse datasets processed using HiCtool.
HiCtool leads the user step-by-step through a pipeline which goes from the source data to the computation, visualization and storage of intra-chromosomal contact maps and topological domain coordinates. A large collection of standardized processed data allows to compare different datasets in a consistent way and it saves time of work to obtain data for additional analyses.
GITAR enables to work with Hi-C data even without any programming or bioinformatic expertise and it is available online at www.genomegitar.org as an open source software.

Running title
GITAR: Genome Interaction Tools and Resources

Keywords
Chromatin interaction; Hi-C data; topological domain; Hi-C processing pipeline; processed Hi-C data library

INTRODUCTION

Genomes are more than linear sequences, with DNA folding-up into elaborate physical structures that allow for extreme spatial compactness of the genetic material and play also an important role in epigenetic regulation. During the past fifteen years, several techniques have been developed to explore the architecture of genomes, such as Chromosome Conformation Capture (3C) [1], 4C [2], 5C [3], Hi-C [4] and ChIA-PET [5]. Among these techniques, Hi-C is one of the most important and it allowed for the first time a genome-wide mapping of chromatin interactions. In order to process Hi-C data several steps are required, including a normalization procedure to remove biases related to the experimental protocol and genomic features. Mostly during the past five years, several software applications have been developed to analyze and visualize genomic interaction data, with different characteristics and outputs. If we consider Hi-C, available analysis software applications are chromoR [6], HiCdat [7], HiCNorm [8], Hi-Corrector [9], Hi-C Pipeline [10], HiC-Pro [11], HiCUP [12], HiFive [13], HIPPIE [14], HiTC [15], HOMER [16], ICE [17]. However, besides correcting biases and generating contact maps, most of them do not provide the entire pipeline to pre-process the raw data (downloadable files) or to compute topological domain coordinates. To call significant chromosome contacts CHiCAGO [18], Fit-Hi-C [19] and HMRFBayesHiC [20] are available, while chromoR and diffHiC [21] can be used to compare spatial interactions between cell lines. For topological domain analysis, HiCseg [22], HubPredictor [23] and TADtree [24] serve to calculate the domain coordinates. If we consider ChIA-PET, ChIA-PET tool [25], Chiasig [26], Mango [27], MDM [28] and MICC [29] can be used to process and analyze data.

Albeit the large number of software, a tool to perform a comprehensive Hi-C data analysis, including pre-processing, normalization, visualization and topological domains computation, was still missing. We have addressed this need by developing HiCtool, a bioinformatic software for Hi-C data analysis, with the aim of creating a standardized, easy and flexible framework to process and visualize Hi-C datasets, and perform a comprehensive intra-chromosomal and topological domains analysis. Moreover, although there are already dozens of source Hi-C datasets released to the public, there is a relatively small number of processed data accessible to researchers to be utilized directly, so we successfully ran and uploaded an exhaustive collection of human and mouse processed datasets, of different cell lines and conditions.

Here we present GITAR (Genome Interaction Tools and Resources), a comprehensive solution to manage Hi-C genomic interaction data, from processing to storage and visualization, composed of the two modules mentioned above: HiCtool and the processed data hub.

RESULTS

HiCtool: a standardized pipeline to process and visualize Hi-C data

HiCtool is an open-source bioinformatic tool based on Python, which integrates several software to perform a standardized Hi-C data analysis, from the source data to the visualization of intra-chromosomal heatmaps [4] and the identification of topological domains [30]. We implemented a pipeline divided into three main sections: data pre-processing, data analysis and visualization, and topological domains analysis (Figure 1).

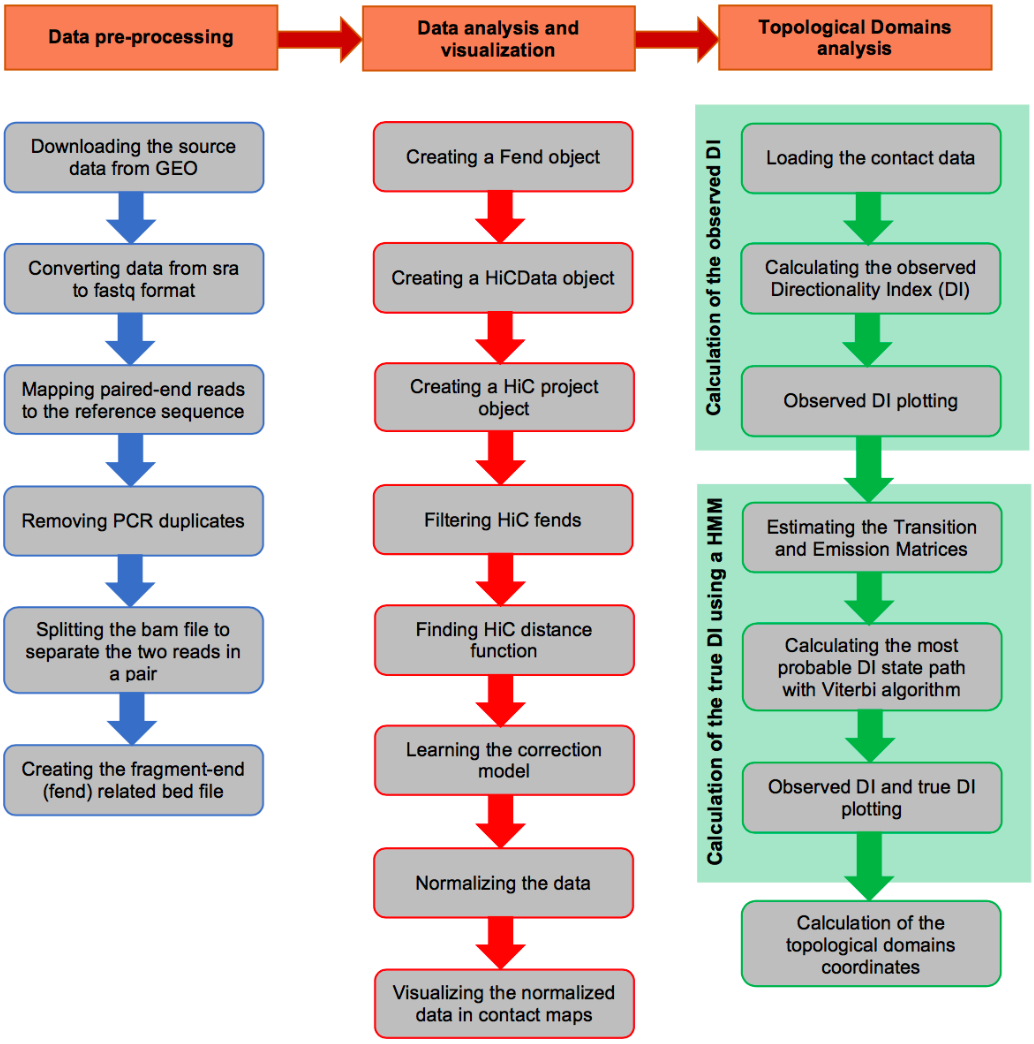
Data pre-processing
Data analysis and visualization
Topological Domains analysis
Downloading the source data from GEO
Converting data from sra to fastq format
Mapping paired-end reads to the reference sequence
Removing PCR duplicates
Splitting the bam file to separate the two reads in a pair
Creating the fragment-end (fend) related bed file
Creating a Fend object
Creating a HiCData object
Creating a HiC project object
Filtering HiC fends
Finding HiC distance function
Learning the correction model
Normalizing the data
Visualizing the normalized data in contact maps
Calculation of the observed DI
Loading the contact data
Calculating the observed Directionality Index (DI)
Observed DI plotting
Calculation of the true DI using a HMM
Estimating the Transition and Emission Matrices
Calculating the most probable DI state path with Viterbi algorithm
Observed DI and true DI plotting
Calculation of the topological domains coordinates

Figure 1. HiCtool workflow. HiCtool is divided into three main sections: data pre-processing (left), data analysis and visualization (center) to normalize contact data and plot heatmaps, and topological domains analysis (right) to calculate topological domain coordinates.

The data pre-processing pipeline takes downloadable files in *sra* format as input and performs several steps to generate input files for the normalization procedure. To do this, HiCtool integrates several software including SRA Toolkit, Bowtie 2, SAMTools and Bedtools. The pre-processing steps include: downloading of the *sra* data; conversion of the data from *sra* to *fastq* format (SRA Toolkit); mapping of the paired-reads over the reference genome (Bowtie 2); removing PCR duplicates from the output *bam* file (SAMTools); splitting the *bam* file into two *bam* files, related to the first and the second read in the pairs; mapping the restriction enzyme (used in the Hi-C experimental protocol [4]) sites over the reference genome to create a fragment-end (fend) related *bed* file. The fend *bed* file is used to normalize the data and it contains restriction sites coordinates and additional information related to the GC content of the fragments. The outputs of the pre-processing section are two *bam* files related to the first and second read in the pairs and the fend *bed* file.

The data analysis and visualization section provides the pipeline to normalize the data and plot the contact heatmaps. The complex experimental Hi-C protocol unavoidably produces several biases. According to Yaffe and Tanay [10], the most significant biases are related to spurious ligation products between fragments, transcription start sites (TSSs) and CTCF-bound sites within topological domains, length and GC content of the fragments. Spurious ligation products generate paired-reads whose sum of the two distances to the nearest restriction sites is larger than 500 bp. About TSSs, analysis of the distribution of *cis* contacts involving fragment ends located 0-5 kb upstream of an active TSS showed a strong enrichment 20 kb to ~400 kb upstream and downstream, increasing the probability that long-range contacts may be associated with the active transcriptional state. In addition, fragments located 0-5 kb on one side of a CTCF-binding site displayed *cis* contacts asymmetry over a range up to ~400 kb. About fragment length, long and short fragments may have a variable ligation efficiency. The probability of contact can be also influenced by the GC content near the ligated fragment ends, up to 200 bp next to the restriction sites. To consider and correct all the mentioned biases, we normalized the data according to Yaffe and Tanay's model [10], performed using the Binning algorithm of the package HiFive [13]. We chose this normalization method because it derives from a comprehensive biological background about Hi-C potential sources of bias and it is one of the most popular approaches. We did not use the also popular HiC Pipeline (hicpipe) [10] because HiFive's Binning algorithm has a more consistent performance across all binning resolutions and, at bin sizes lower than 50 kb, HiFive's performance is better [13]. This is a crucial point for our pipeline since as default we processed the data at a resolution of 40 kb to enable the topological domains analysis [30]. In addition, HiFive's capability of handling high-resolution data makes it able to process the last generation of Hi-C datasets, created with the "in situ Hi-C" protocol [31]. This derives from the use of the HDF5 binary data format, which allows to store a huge amount of data with a high efficient memory usage. Lastly, about the running time, HiFive's Binning algorithm performs faster not only than hicpipe, but also than the other normalization tools based on R, which is slower compared to Python.

According to the biases listed above, in our pipeline we removed the paired reads whose total distance from the nearest restriction sites was greater than 500 bp. Then, we filtered out fragments interacting within a distance of 500 kb before learning the correction parameters related to fend biases (length and GC content). This allowed to eliminate the effects of biased regions upstream or downstream of an active TSS or CTCF-binding site. To remove the bias related to the GC content, we added the information about the GC content of regions up to 200 bp next to the restriction sites to the fend related *bed* file. After that, correction parameters for fragment length and GC content were learned using Yaffe and Tanay's method [10], dividing the length and GC content ranges into 20 bins, such that each bin contained the same number of fragments. For the optimization process of the correction matrices by likelihood maximization, we used a learning threshold of 1 (the same of Yaffe and Tanay) and a maximum number of iterations of 1000. At this point, for any arbitrary division of the genome into bins, we computed two matrices for each chromosome: an observed intra-chromosomal contact matrix *O[i,j]*, where each entry contains the observed reads count between the regions identified by the bins *i* and *j*, and a "fend" expected contact matrix *E[i,j]*, where each entry contains the sum of corrections for all the paired-reads between bins *i* and *j*. Then, the normalized contact matrix *N[i,j]* is calculated according to:

$$N[i,j] = \frac{O[i,j]}{E[i,j]}$$

where each entry contains the corrected reads count according to the previous model. We computed also an "enrichment" expected contact matrix, which represents the expected read counts considering the learned correction parameters and the distance between fends. In particular, the average intrachromosomal contact probability for pairs of loci decreases monotonically with increasing of the linear genomic distance [4]. The enrichment value was then calculated as the ratio between the observed and the "enrichment" expected data. The pipeline allows finally to plot the contact heatmaps, with additional colorbar and histogram of the output data (Figure 2).

The topological domains analysis section provides the code to calculate the Directionality Index (DI) and topological domains coordinates. It computes either the observed DI and the hidden "true" DI using a Hidden Markov Model (HMM). Both the observed and the HMM biased states ("true" DI) can be plotted in the same figure, therefore it is possible to infer easily about the presence of topological domains and boundaries over the genome (Figure 3). Topological domain coordinates are then calculated using the shifts of the biased HMM states according to Dixon et al. [30].

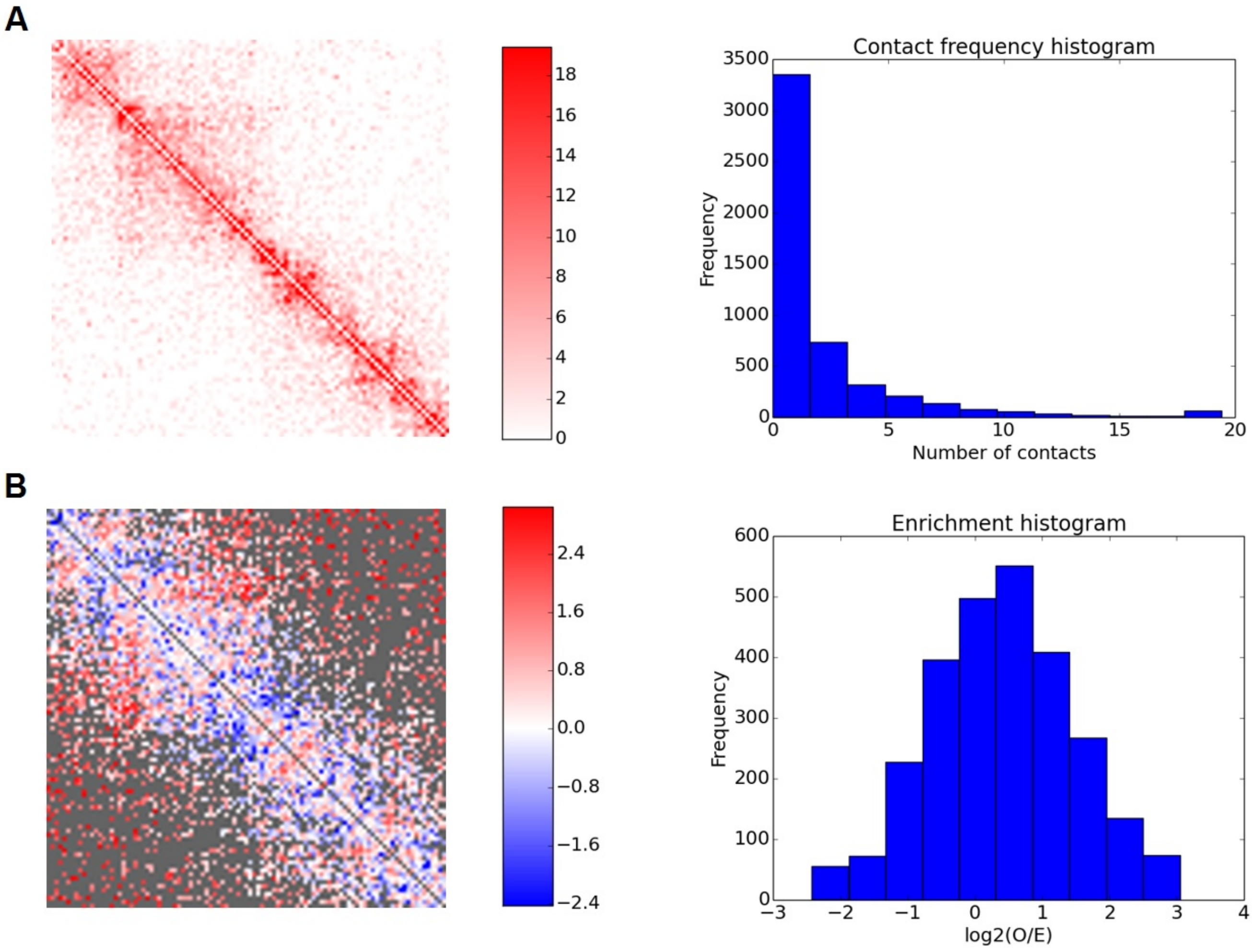

Figure 2. Chr 6 [50-54] Mb at a bin size of 40 kb: contact data. (A) Normalized contact matrix where each bin contains the corrected read counts. Range from 0 to 19 reads. **(B)** Enrichment normalized contact matrix where each bin contains the $\log_2$ of the enrichment value (observed over expected data considering the distance between fends and the learned correction parameters). The gray pixels represent non-valid $\log_2$ (enrichment) values, where the corresponding expected value is 0. Range from -2.421 to 3.047. GEO accession number: GSM862723.

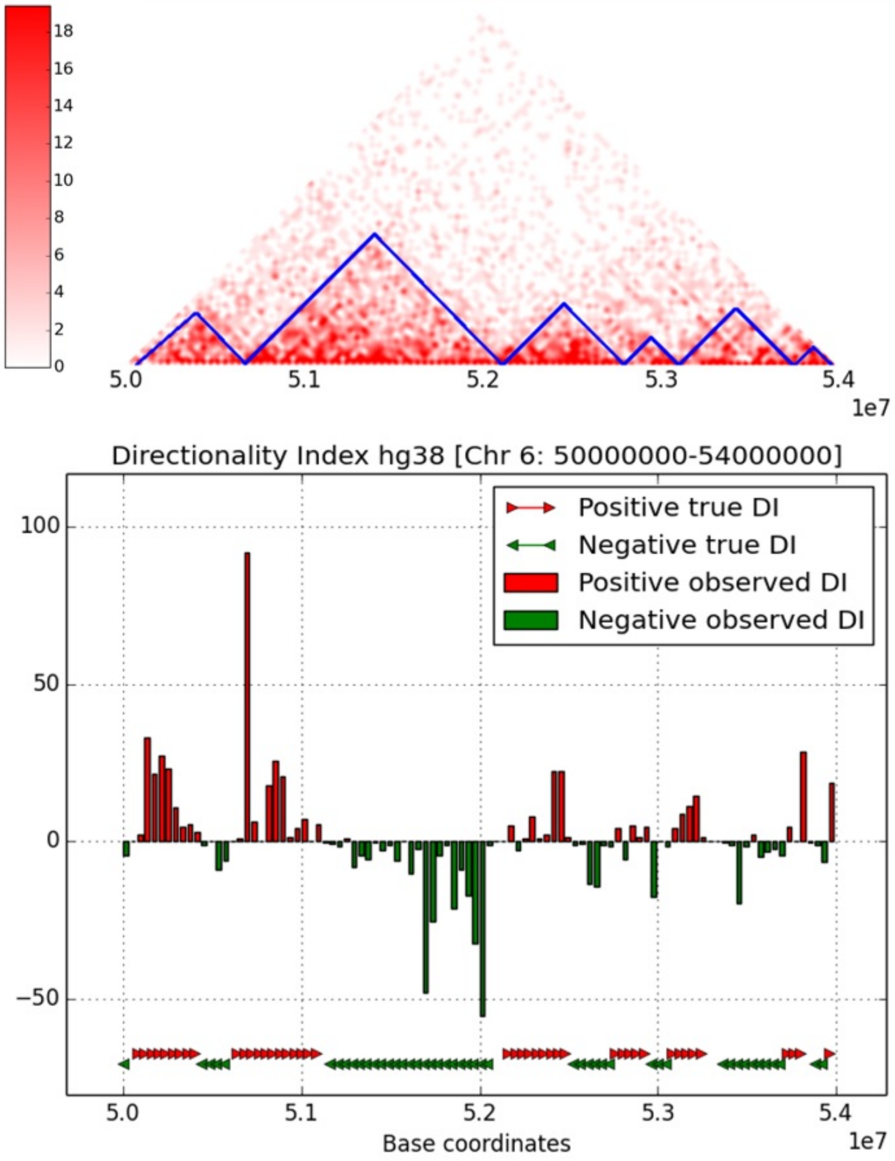

Figure 3. Chr 6 [50-54] Mb at a bin size of 40 kb: observed DI and HMM biased states. The plot shows the observed DI values (barplot) and the HMM biased states (arrows). Positive true DI and negative true DI in the legend refer to downstream and upstream biased states respectively. According to the HMM state shifts, six topological domains and seven topological domain boundaries are shown. Topological domains show up as triangles along the diagonal of the heatmap and they are highlighted in blue. Here the heatmap is represented only as the triangular part laying on the diagonal. GEO accession number: GSM862723.

Processed data module

The processed data module is a collection of standardized processed datasets using HiCtool. As shown in Table 1, we have already run 19 datasets of Homo Sapiens (hg38 reference genome), taken from the library on the 4D Nucleome Web Portal (www.4dnucl.org) [32], and 2 datasets of Mus Musculus (mm10 reference genome). The 4DN library is a collection of genome interaction papers related to the Chromosome Conformation Capture based assays (3C, 4C, 5C and Hi-C). Specifically, here we referred only to Hi-C derived datasets.

Four different outputs for each chromosome of a processed dataset are computed and saved with HiCtool: contact matrices, DI values, HMM biased states (“true” DI) and topological domain coordinates. For contact matrices, observed contacts, expected contacts (fend, enrichment) and normalized contacts (fend, enrichment) are computed. All the outputs are in *txt* format, and the functions to save and load the data are given. To reduce storage usage and computing time, we used an efficient data format to store contact matrices, exploiting that they are symmetric and sparse (see Materials and Methods). For the normalized contact data of the human cell line hESC line H1 (GEO accession number: GSM862723) at the resolution of 40 kb, ~1.3 GB of storage is required if the full contact data are saved, only ~97 MB using our method.

This means that storage usage is reduced of ~92% (~58% if the files are compressed). About the computing time, saving and loading full contact matrices with 16 GB of RAM require respectively ~6 minutes and ~3 minutes. Saving and loading our parsed data require 1 minute for both. This means ~83% and ~67% of data saving and loading time reduction respectively.

Conversely, domain coordinates are in a format to be read directly. Each line of the *txt* file refers to a topological domain, with tab separated start and end coordinates, allowing easy access and readability.

GEO Accession Number	Restriction enzyme	Species	Cell Line	Reference
GSM455133	HindIII	Homo Sapiens	EBV-transformed lymphoblastoid GM06990	[4]
GSM862723	HindIII	Homo Sapiens	Human Embryonic Stem Cells H1	[30]
GSM862724	HindIII	Homo Sapiens	Fetal lung fibroblast IMR90	[30]
GSM862720	HindIII	Mus Musculus	Mouse Embryonic Stem Cells J1	[30]
GSM1551633	MboI	Mus Musculus	B-lymphoblasts CH12-LX	[31]
GSM1551550	MboI	Homo Sapiens	B-lymphoblastoids GM12878	[31]
GSM1551599	MboI	Homo Sapiens	Lung Fibroblast IMR90 (CCL-186)	[31]
GSM927075	HindIII	Homo Sapiens	ERG prostate epithelial cell line RWPE-1	[33]
GSM1055800	HindIII	Homo Sapiens	Fetal lung fibroblast IMR90	[34]
GSM1055805	HindIII	Homo Sapiens	Embryonic stem cells H1	[34]
GSM1906332	HindIII	Homo Sapiens	B-cell Follicular Lymphoma RL	[35]
GSM1906333	HindIII	Homo Sapiens	Primary TumorB-cell acute lymphocytic leukemia B-ALL	[35]
GSM1906334	HindIII	Homo Sapiens	MHH-CALL- 4 B-cell acute lymphocytic leukemia CALL4	[35]
GSM1250485	HindIII	Homo Sapiens	Breast cancer cells MCF-7	[36]
GSM1294038	HindIII	Homo Sapiens	Breast cancer cell line T47D-MTVL, unstimulated	[37]
GSM1294039	HindIII	Homo Sapiens	Breast cancer cell line T47D-MTVL, progestin R5020-stimulated	[37]
GSM1267200	HindIII	Homo Sapiens	Mesenchymal stem cells H1	[38]
GSM1267196	HindIII	Homo Sapiens	Embryonic stem cells H1	[38]
GSM1718021	HindIII	Homo Sapiens	Human Embryonic Stem Cells H9	[39]
GSM1909121	HindIII	Homo Sapiens	Haploid fibroblast-like Hap1	[40]
GSM1081530	HindIII	Homo Sapiens	HEK293T	[41]

Table 1. GITAR processed data. Datasets processed using HiCtool and available for downloading at data.genomegitar.org.

DISCUSSION

As we presented above, GITAR is a comprehensive tool, which consists of a pipeline to analyze genomic interaction data (HiCtool) and a large collection of processed datasets. HiCtool establishes a standardized, flexible and easy way to work with Hi-C data, and it integrates all the software needed for the analysis. This allows users to work on different datasets and make comparisons in a consistent way. For each section, a tutorial leads through each of the analysis steps, making the entire procedure simple, clear and user-friendly, with the key advantage that no programming expertise or additional software documentation are required. Besides the tutorial part, the API documentation in each section shows the syntax of all the functions that are used. In such a way, the main difference with the other available packages is that HiCtool is a comprehensive solution which integrates Hi-C data pre-processing, normalization, visualization and topological domains computation, instead of focusing only on a part of the analysis. Having one comprehensive tool has the key advantage of avoiding the difficulty related to data integration, when different tasks are performed by different packages and the input data needs to be supplied in the format required by each specific software. In addition, HiCtool was built with a pipeline structure to make it easy to use, flexible and suitable to every kind of user. This gives full control over the analysis instead of generating output of processes that especially beginners would not be able to clearly understand or manage.

We built also a large library, to provide a collection of standardized processed datasets from several cell lines ready for downloading. Contact matrices, Directionality Index, HMM biased states and topological domain coordinates are available for each cell line. Since contact matrices may be really big, especially with the increasing of the resolution, they are saved in a parsed format to reduce memory usage and allow quick data access even with a personal laptop computer. Having a collection of datasets processed with the same workflow is crucial to perform consistent comparisons and it provides data directly ready for further analyses.

MATERIALS AND METHODS

HiCtool is an open source software and the source code is available for downloading at doc.genomegitar.org. The pre-processing (Figure 1, left) is based on several applications (SRA Toolkit, Bowtie 2, SAMTools and Bedtools), hence it consists of a pipeline where each command is performed through UNIX lines of code. The rest of the analysis (Figure 1, center and right) is based on Python functions, therefore tasks can be performed with simple function calls after running the scripts.

Data normalization and heatmaps

Contact data are normalized using the Binning algorithm of the Python package HiFive (v1.2.1) [13]. Heatmaps are plotted using the Python Image Library (PIL). Colorbar and histogram are generated using the Python libraries Matplotlib and Matplotlib.pyplot.

Directionality Index and Hidden Markov Model

Given a division of the genome into 40 kb bins, we quantified the observed Directionality Index (DI) using the following formula from Dixon et al. [30]:

$$DI = \left(\frac{B-A}{|B-A|}\right)\left(\frac{(A-E)^2}{E}+\frac{(B-E)^2}{E}\right)$$

where A is the number of reads that map from a given 40 kb bin to the upstream 2 Mb region, B is the number of reads that map from a given 40 kb bin to the downstream 2 Mb region and E is the expected number of contacts for each bin and it equals to $\frac{\mathrm{A+B}}{2}$. Therefore, the fend normalized contact data at a bin size of 40 kb is needed to compute the DI. The detection region of 2 Mb for upstream or downstream biases corresponds to 50 bins (2 Mb / 40 kb = 50 bins).

We used a Hidden Markov Model (HMM) based on the Directionality Index to identify biased states. To perform the HMM we used the Python package hmmlearn. Specifically, we built a model with three biased states corresponding to downstream bias, upstream bias and no bias. The sequence of emissions corresponds to the observed DI values and transition matrix, emission matrix and initial state sequence are unknown. We have three types of emissions named as 1, 2, 0 in the model and corresponding to a positive (1), negative (2) or zero (0) value of the observed DI. In our analysis, we associated to the emission ‘0’ all the absolute DI values under a threshold of 0.4. We initialized transition and emission matrices with the same values of 0.3 for the probabilities to transit to a different state or emission respectively (values outside the diagonal), and 0.4 for the probabilities of remaining in the same state or observing the same emission (values in the diagonal). So, first we estimated the model parameters and then the most probable sequence of states using the Viterbi algorithm. Biased states were then exploited to calculate topological domain coordinates. According to Dixon et al. [30], a domain is initiated at the beginning of a single downstream biased state. The domain is continuous throughout any consecutive downstream biased states and ends when the last in a series of upstream biased states is reached.

Contact map storage

HiCtool allows to generate and save contact matrices at the resolution defined by the user. In our pipeline, we processed Hi-C data with a bin size of 40 kb to allow topological domains analysis according to Dixon et al. [30]. Already at this resolution, contact matrices contain several million of elements per each chromosome, requiring big storage space and relatively high data saving and loading time. To address this problem, we proposed a way to parse the data based on the fact that contact maps are symmetric (contacts between loci *i* and *j* are the same than those between loci *j* and *i*) and usually sparse, since most of the elements are zeros, and this property is stronger with the decreasing of the bin size. Given these two properties, it is not needed to save mirrored data and moreover it would be useful to “compress” the zero data within the matrices. To accomplish this, first we selected only the upper-triangular part of the contact matrices (including the diagonal) and we reshaped the data by rows to form a vector. After that, we replaced all the consecutive zeros in the vector with a “0” followed by the number of zeros that are repeated consecutively; all the non-zero elements are left as they are. Finally, the data are saved in a *txt* file (see Figure 4). As mentioned before, at higher resolutions (low bin sizes) contact matrices show more zeros, meaning that the advantage given by this data format would be more remarkable in terms of storage usage and computation time.

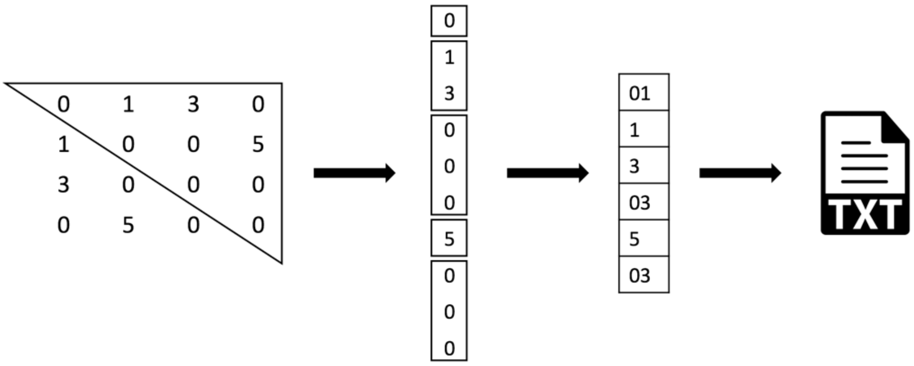

Figure 4. Contact map storage workflow. This is a simplified example where an intra-chromosomal contact matrix is represented by a 4x4 symmetric and sparse matrix. 1) The upper-triangular part of the matrix is selected (including the diagonal). 2) Data are reshaped to form a vector. 3) All the consecutive zeros are replaced with a “0” followed by the number of zeros that are repeated consecutively. 4) Data are saved into a *txt* file.

Software requirements

This is the list of software that are required to use GITAR: Python (>2.7), Bowtie 2, Bedtools, SAMTools, SRA Toolkit. The Python libraries needed are: Numpy, Scipy, Math, Matplotlib, Matplotlib.pyplot, PIL, csv. Additional Python packages are: HiFive and hmmlearn. HiFive is used to normalize contact data, while hmmlearn serves for the Hidden Markov Model to calculate the biased states used to extract topological domain coordinates.

ACKNOWLEDGEMENT

This work is funded by NIH U01CA200147 and DP1HD087990.

REFERENCES

[1] Dekker J., Rippe K., Dekker M., & Kleckner N. (2002). Capturing chromosome conformation. Science, 295(5558), 1306-1311.
[2] Zhao Z., Tavoosidana G., Sjölinder M., Göndör A., Mariano P., Wang S., et al. (2006). Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra-and interchromosomal interactions. *Nature genetics*, *38*(11), 1341-1347.
[3] Dostie J., Richmond T. A., Arnaout R. A., Selzer R. R., Lee W. L., Honan T. A., et al. (2006). Chromosome Conformation Capture Carbon Copy (5C): a massively parallel solution for mapping interactions between genomic elements. *Genome research*, *16*(10), 1299-1309.
[4] Lieberman-Aiden E., Van Berkum N. L., Williams L., Imakaev M., Ragoczy T., Telling A., et al. (2009). Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *science*, *326*(5950), 289-293.
[5] Zhang J., Poh H. M., Peh S. Q., Sia Y. Y., Li G., Mulawadi F. H., et al. (2012). ChIA-PET analysis of transcriptional chromatin interactions. *Methods*, *58*(3), 289-299.
[6] Shavit Y. "Combining a wavelet change point and the Bayes factor for analysing chromosomal interaction data." *Molecular BioSystems* 10.6 (2014): 1576-1585.
[7] Schmid M. W., Grob S, Grossniklaus U. "HiCdat: a fast and easy-to-use Hi-C data analysis tool." *BMC bioinformatics* 16.1 (2015): 277.
[8] Hu M., Deng K., Selvaraj S., Qin Z. S., Ren B., Liu J. S. (2012). HiCNorm: removing biases in Hi-C data via Poisson regression. *Bioinformatics* 28 (23), 3131-3133.
[9] Li W., Gong K., Li Q., Alber F., Zhou X. J. (2014). Hi-Corrector: a fast, scalable and memory-efficient package for normalizing large-scale Hi-C data. *Bioinformatics*, *31*(6), 960-962.
[10] Yaffe E., Tanay A. "Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture."*Nature genetics* 43.11 (2011): 1059-1065.
[11] Servant N., Varoquaux N., Lajoie B. R., Viara E., Chen C. J., Vert J. P., et al. (2015). HiC-Pro: an optimized and flexible pipeline for Hi-C data processing. *Genome biology*, *16*(1), 259.
[12] Wingett S., Ewels P., Furlan-Magaril M., Nagano T., Schoenfelder S., Fraser P., Andrews S. (2015). HiCUP: pipeline for mapping and processing Hi-C data. *F1000Research*, *4*.
[13] Sauria M. E., Phillips-Cremins J. E., Corces V. G., Taylor J. (2015). HiFive: a tool suite for easy and efficient HiC and 5C data analysis. *Genome biology*, *16*(1), 237.
[14] Hwang Y. C., Lin C. F., Valladares O., Malamon J., Kuksa P. P., Zheng Q., et al. (2014). HIPPIE: a high-throughput identification pipeline for promoter interacting enhancer elements. *Bioinformatics*, *31*(8), 1290-1292.
[15] Servant N., Lajoie B. R., Nora E. P., Giorgetti L., Chen C. J., Heard E., et al. (2012). HiTC: exploration of high-throughput 'C'experiments. *Bioinformatics*, *28*(21), 2843-2844.
[16] Heinz S., Benner C., Spann N., Bertolino E., Lin Y. C., Laslo P., et al. (2010). Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Molecular cell*, *38*(4), 576-589.
[17] Imakaev M., Fudenberg G., McCord R. P., Naumova N., Goloborodko A., Lajoie B. R., et al. (2012). Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nature methods*, *9*(10), 999-1003.
[18] Cairns J., Freire-Pritchett P., Wingett S. W., Várnai C., Dimond A., Plagnol V., et al. (2016). CHiCAGO: robust detection of DNA looping interactions in Capture Hi-C data. *Genome biology*, *17*(1), 127.
[19] Ay F., Bailey T. L., Noble W. S. (2014). Statistical confidence estimation for Hi-C data reveals regulatory chromatin contacts. *Genome research*, *24*(6), 999-1011.
[20] Xu Z., Zhang G., Jin F., Chen M., Furey T. S., Sullivan P. F., et al. (2015). A hidden Markov random field-based Bayesian method for the detection of long-range chromosomal interactions in Hi-C data. *Bioinformatics*, *32*(5), 650-656.
[21] Lun A. T., Smyth G. K. (2015). diffHic: a Bioconductor package to detect differential genomic interactions in Hi-C data. *BMC bioinformatics*, *16*(1), 258.
[22] Lévy-Leduc C., Delattre M., Mary-Huard T., Robin S. (2014). Two-dimensional segmentation for analyzing Hi-C data. *Bioinformatics*, *30*(17), i386-i392.
[23] Huang J., Marco E., Pinello L., Yuan G. C. (2015). Predicting chromatin organization using histone marks. *Genome biology*, *16*(1), 162.

[24] Weinreb C., Raphael B. J. (2015). Identification of hierarchical chromatin domains. *Bioinformatics*, *32*(11), 1601-1609.
[25] Li G., Fullwood M. J., Xu H., Mulawadi F. H., Velkov S., Vega V., et al. (2010). ChIA-PET tool for comprehensive chromatin interaction analysis with paired-end tag sequencing. *Genome biology*, *11*(2), R22.
[26] Paulsen J., Rødland E. A., Holden L., Holden M., Hovig E. (2014). A statistical model of ChIA-PET data for accurate detection of chromatin 3D interactions. *Nucleic acids research*, *42*(18), e143-e143.
[27] Phanstiel D. H., Boyle A. P., Heidari N., Snyder M. P. (2015). Mango: a bias-correcting ChIA-PET analysis pipeline. *Bioinformatics*, *31*(19), 3092-3098.
[28] Niu L., Lin S. (2015). A Bayesian mixture model for chromatin interaction data. *Statistical applications in genetics and molecular biology*, *14*(1), 53-64.
[29] He C., Zhang M. Q., Wang X. (2015). MICC: an R package for identifying chromatin interactions from ChIA-PET data. *Bioinformatics*, *31*(23), 3832-3834.
[30] Dixon J. R., Selvaraj S., Yue F., Kim A., Li Y., Shen Y., et al. (2012). Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, *485*(7398), 376-380.
[31] Rao S. S., Huntley M. H., Durand N. C., Stamenova E. K., Bochkov I. D., Robinson J. T., et al. (2014). A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, *159*(7), 1665-1680.
[32] Dekker J., Belmont A. S., Guttman M., Leshyk V. O., Lis J. T., Lomvardas S., et al. (2017). The 4D nucleome project. *bioRxiv*, 103499.
[33] Rickman D. S., Soong T. D., Moss B., Mosquera J. M., Dlabal J., Terry S., et al. (2012). Oncogene-mediated alterations in chromatin conformation. *Proceedings of the National Academy of Sciences*, *109*(23), 9083-9088.
[34] Jin F., Li Y., Dixon J. R., Selvaraj S., Ye Z., Lee A. Y., et al. (2013). A high-resolution map of the three-dimensional chromatin interactome in human cells. *Nature*, *503*(7475), 290-294.
[35] Wang Z., Cao R., Taylor K., Briley A., Caldwell C., Cheng J. (2013). The properties of genome conformation and spatial gene interaction and regulation networks of normal and malignant human cell types. *PloS one*, *8*(3), e58793.
[36] Mourad R., Hsu P. Y., Juan L., Shen C., Koneru P., Lin H., et al. (2014). Estrogen induces global reorganization of chromatin structure in human breast cancer cells. *PLoS One*, *9*(12), e113354.
[37] Le Dily F., Baù D., Pohl A., Vicent G. P., Serra F., Soronellas D., et al. (2014). Distinct structural transitions of chromatin topological domains correlate with coordinated hormone-induced gene regulation. *Genes & development*, *28*(19), 2151-2162.
[38] Dixon J. R., Jung I., Selvaraj S., Shen Y., Antosiewicz-Bourget J. E., Lee A. Y., et al. (2015). Chromatin architecture reorganization during stem cell differentiation. *Nature*, *518*(7539), 331-336.
[39] Nagano T., Várnai C., Schoenfelder S., Javierre B. M., Wingett S. W., Fraser P. (2015). Comparison of Hi-C results using in-solution versus in-nucleus ligation. *Genome biology*, *16*(1), 175.
[40] Sanborn A. L., Rao S. S., Huang S. C., Durand N. C., Huntley M. H., Jewett A. I., et al. (2015). Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes. *Proceedings of the National Academy of Sciences*, *112*(47), E6456-E6465.
[41] Zuin J., Dixon J. R., van der Reijden M. I., Ye Z., Kolovos P., Brouwer R. W., et al. (2014). Cohesin and CTCF differentially affect chromatin architecture and gene expression in human cells. *Proceedings of the National Academy of Sciences*, *111*(3), 996-1001.