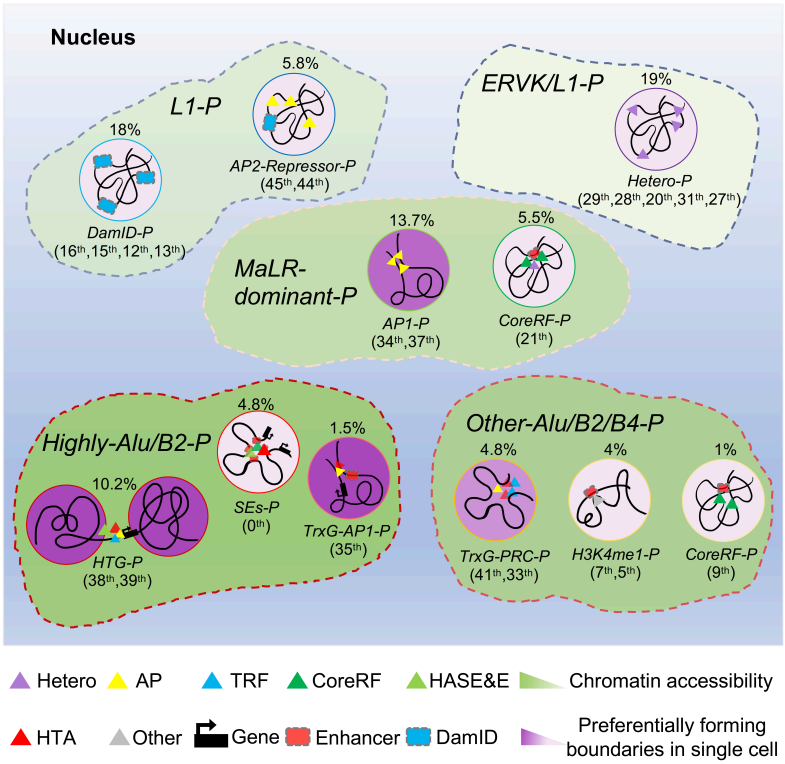
Deciphering Hierarchical Chromatin Structures and Preference of Genomic Positions in Single Mouse Embryonic Stem Cells



In Brief

Multiple types of regulatory factors interplaying with each other in specific genomic positions could affect focal chromatin interactions, thereby changing interaction density or insulation strength of these regions, which may navigate the preference of genomic positions for forming boundaries, shape hierarchical chromosome architecture, and then regulate gene expression and cell functions in single embryonic stem cells.

Highlights

- Developing **HiCS** to detect hierarchical chromatin structures from single-cell Hi-C maps
- Reorganizing an atlas of ChIP-seq for mouse embryonic stem cells (mESCs) containing hundreds of regulatory factors, which are either enriched or not in domain boundaries of single cells with several different enrichment patterns
- Grouping and annotating spatially organized chromatin landscape clusters with regulatory factors, providing an increasingly complex view of the genomic structure-function relationship
- The above chromatin landscape-clusters clearly correspond to five large functional units based on retrotransposons and potentially regulate function-specific cellular states.

Keywords

Hierarchical chromatin structure, single-cell Hi-C map, regulatory factor, chromatin landscape, retrotransposon

HiCS: a method for the identification of hierarchical chromatin structures from single-cell Hi-C data

Contents:

Introduction:

The key design of the HiCS algorithm is to convert the problem of the identification of hierarchical chromatin structures into finding peaks of insulation strength at different genome scales. We observe that the domain boundaries have higher insulation strength than their neighbors and a relatively large distance from any regions with higher strength (Fig. 1e). This process only calculates the two metrics (insulation strength ρ and the minimum distance between the bin and any other bin with higher strength δ) for each bin, and can intuitively control the number of peaks to obtain the hierarchical chromatin structures at different scales by α (Fig1b-1c). Thus, the algorithm is super-fast to identify a chromatin hierarchy (i.e., the structure of the higher level embraces the multiple structures of the lower level in Fig. 1d-1e). Noting that high-level boundaries with high δ may have lower insulation strength than low-level boundaries, and boundaries in the same level have magnitude differences in local insulation strength (Fig. 1e).

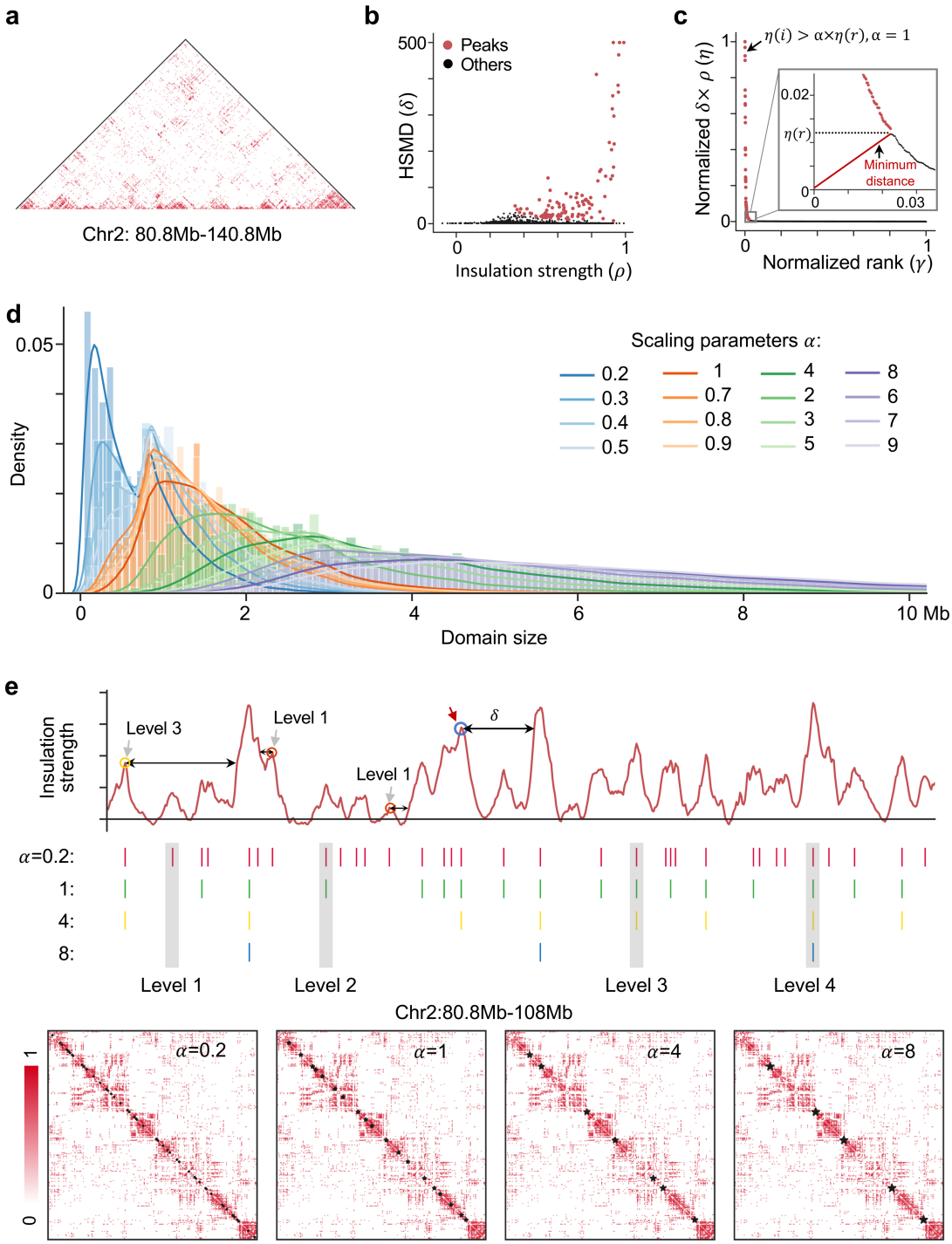


Fig. 1. The overview of HiCS.

Installation:

You can download source code of HiCS here, and the package requires the following prerequisites: HiCS is developed under python (version >= 3.6), and HiCS also requires the installation of node2vec==0.4.2, networkx==2.5.1, pandas==1.2.3, numpy==1.20.2, h5py==3.2.1, scikit-learn==0.24.1 and scipy==1.6.2 etc.

Runing HiCS for an example:

Step 0. The following input files are mandatory for HiCS.

1. chrom.sizes file for the genome of interest, which can be downloaded from here. Files for mm9, mm10, hg19, and hg38 are included in the ext directory.

2. Put all the contact files, one for each cell, into the same directory ('Datasets_Single_mESCs'). In each contact (read pairs) file of single cells, one line represents one contact with 2 columns for chromosome name and 2 columns for the mapped positions (bp). For example, lines of '.adj' file are shown in the 'Datasets_Single_mESCs' directory.

Step 1. Binning and imputing the contact files of single cells.

0. Firstly, counting the read pairs for every single cell to save 'ext/contact_num.csv';
1. Secondly, each chromosome first is divided into specific size bins (40kb in the example), and contacts are counted for each bin pair. The binned files are saved to 'Datasets_Single_mESCs_40kb/binned';
2. Thirdly, we model each chromosome as an unweighted network (each bin is one node, and each bin pair with non-zero contacts is added one edge), and implement a classic graph embedding method node2vec, which applies a biased random walks procedure, to compute contact probability of edges by computing the cosine similarity of any two node embedding vectors, and obtain imputed matrix. The imputed files are saved to 'Datasets_Single_mESCs_40kb/rl_csc_95';

The above steps are performed with the following command：

Python Run_Step0-1_Single_mESC.py

Step 2. Detecting hierarchical chromatin structures of single cells.

1. The imputing file of each chromosome is inputted to the step. The identification of chromatin structures of single cells is performed with the following command:

   Python Run_Step2_Single_mESC.py

2. Setting different parameters "SCALE" in 'Run_Step2_Single_mESC.py' to generate domain boundaries in different genomic scales.

Step 3. Output files.

The results files in SCALE=1 are saved to 'Datasets_Single_mESCs_40kb/sctadboundaries_1'.

Parameters setting:

Before you run a new dataset, you need to change the corresponding parameters in Run_Step0-1_Single_mESC.py or Run_Step2_Single_mESC.py or set new user variables according to your situation. Alternatively, you can run HiCS using multithreaded and single-processor environments, provided enough memory. The main parameters are listed in the following:

1. package_dir: the directory of HiCS package.
2. -i --indir: input directory for contact files of single cells, which is specified by INDIR in the above example.
3. -s --suffix: suffix of contact files of single cells. The default value is '.adj'.
4. -o --outdir: output directory, which is specified by OUTDIR in the above example.
5. -c --chr-columns: the location of two columns for chromosome in the above contact files. The default value is [0, 3].
6. -p --pos-columns: the location of two columns for read positions in the above contact files. The default value is [1,4].
7. -g --genome: The genome name: 'mmX' or 'hgX'. The default value is 'mm9'.
8. -w --window: The slide window size of computing insulation strength of boundaries. The default value is 20.
9. --scale: Scale parameter for hierarchical chromatin structures. The default value is 1.
10. --binsize: Bin size used for binning the reads. The default value is 4e4
11. –alpha: Reserving the top edges for downstream analysis. The default value is 0.05.
12. --step: setting steps to run. The default is all steps.
13. --threaded: if set True, the HiCS is performed using multiprocessing on a single machine.
14. -n –num-proc: The number of processes used in threaded mode.

There are also several other parameters described in HiCS_main.py/main() with details. Thus, you also can perform your task as the script below.

python HiCS_main.py -i <input-directory> -o <output-directory> -l <chromosome-length-file> -g <genome: hg-/mm-> --binsize <the size of bin bp> --scale <the scale parameter> --step "bin rl sctadboundaries"

Contact us:

For any questions, comments, and suggestions regarding HiCS, please submit an issue to Github with the details of your system and run. You can also send an email to Yusen Ye(ysye@xidian.edu.cn).

Citation:

If you use **HiCS**, please cite our paper.