Bioinformatics, 37(23), 2021, 4562–4563
https://doi.org/10.1093/bioinformatics/btab698
Advance Access Publication Date: 8 October 2021
Applications Note

Genome analysis

iNucs: inter-nucleosome interactions

Mehrdad Oveisi [1,†], Manu Shukla[2,†], Nogayhan Seymen[1], Masae Ohno[3,4], Yuichi Taniguchi [3,4], Sunil Nahata[5], Remco Loos[6], Ghulam J. Mufti[1], Robin C. Allshire[2], Stefan Dimitrov[5,7] and Mohammad M. Karimi [1,*]

[1]Comprehensive Cancer Centre, School of Cancer & Pharmaceutical Sciences, Faculty of Life Sciences & Medicine, King's College London, London SE5 8AF, UK, [2]Wellcome Centre for Cell Biology, Institute of Cell Biology, School of Biological Sciences, University of Edinburgh, Edinburgh EH9 3BF, UK, [3]Institute for Integrated Cell-Material Sciences (iCeMS), Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan, [4]Laboratory for Cell Systems Control, RIKEN Center for Biosystems Dynamics Research, Suita, Osaka 565-0874, Japan[5]Institute for Advanced Biosciences, Inserm U 1209, CNRS UMR 5309, Université Grenoble Alpes, 38000 Grenoble, France, [6]BMS Center for Innovation and Translational Research Europe (CITRE, a Bristol Myers Squibb Company), Pabellón de Italia, 41092 Sevilla, Spain and [7]Izmir Biomedicine and Genome Center, Dokuz Eylul University Health Campus, Balçova, Izmir 35330, Turkey

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Pier Luigi Martelli

Received on April 23, 2021; revised on September 29, 2021; editorial decision on October 4, 2021; accepted on October 5, 2021

Abstract

Motivation: Deciphering nucleosome–nucleosome interactions is an important step toward mesoscale description of chromatin organization but computational tools to perform such analyses are not publicly available.

Results: We developed iNucs, a user-friendly and efficient Python-based bioinformatics tool to compute and visualize nucleosome-resolved interactions using standard pairs format input generated from pairtools.

Availabilityand implementation: https://github.com/Karimi-Lab/inucs/.

Contact: mohammad.karimi@kcl.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

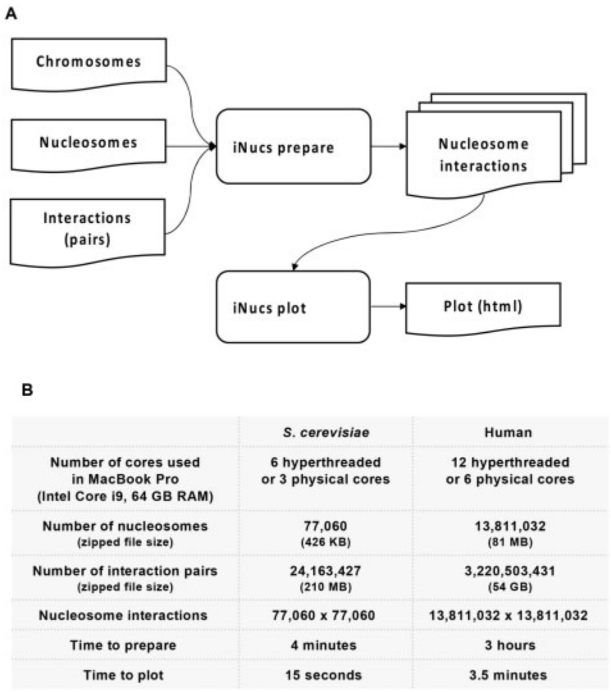
1 Introduction

In eukaryotes, DNA is packaged in the nucleus as a nucleoprotein complex called chromatin. The nucleosome is the fundamental unit of chromatin, around which 147 bp of DNA is wound. The packaging of chromatin in the nucleus is not linear. This results in different levels of 3-D genome folding and interaction. Deciphering genome organization within the nucleus is a multi-scale problem and while higher-order tertiary genome structures have been extensively studied using chromatin conformation capture technologies (e.g. 3C, Hi-C etc.), the secondary structure of chromatin, i.e. nucleosomal interactions, remains elusive. Recently, new genome-wide chromatin conformation approaches have been introduced, allowing interactions at nucleosome level. However, the commonly utilized visualization tools such as HiGlass and Juicebox (Kerpedjiev *et al.*, 2018; Robinson *et al.*, 2018) are limited in their ability to provide nucleosome-resolved interactions due to their underlying bin-based methods that do not take into account the nucleosome positions (Supplementary Fig. S1). Therefore, a dedicated bioinformatics tool which allows analysis and visualization of nucleosome interaction networks at individual loci is needed. In this study, we describe 'iNucs', a user-friendly and efficient Python-based tool for computing and visualizing nucleosome-level interaction networks. iNucs is designed to receive ligation junctions in pairs format generated by pairtools (https://pairtools.readthedocs.io/en/latest/) and provide nucleosome resolution view consistent with pairsQC (https://github.com/4dn-dcic/pairsqc) output and 4D nucleosome (4DN)-processed files. This allows iNucs to be integrated as a plugin to 4DN analysis pipeline in the future. In addition, the tool provides division and visualization of interaction pairs based on their orientation providing further spatial information about the conformation of chromatin fiber at genomic loci.

2 Materials and methods

iNucs consists of prepare and plot modules for which the processes, input and output files and formats are shown in Figure 1A. For a detailed, step-by-step examples of this tool, please see iNucs GitHub page.

Prepare module: This module takes in nucleosome genomic coordinates, ligation junctions (in pairs format) and chromosome names as input, and it produces a nucleosome interaction matrix (NIM) as output. The pairs' input data are typically very large, particularly for human samples (Fig. 1B). Thus, the running time of the underlying



	S. cerevisiae	Human
Number of cores used in MacBook Pro (Intel Core i9, 64 GB RAM)	6 hyperthreaded or 3 physical cores	12 hyperthreaded or 6 physical cores
Number of nucleosomes (zipped file size)	77,060 (426 KB)	13,811,032 (81 MB)
Number of interaction pairs (zipped file size)	24,163,427 (210 MB)	3,220,503,431 (54 GB)
Nucleosome interactions	77,060 x 77,060	13,811,032 x 13,811,032
Time to prepare	4 minutes	3 hours
Time to plot	15 seconds	3.5 minutes

Fig. 1. iNUcs overview and performance. (**A**) Schematics of input data requirements, analysis modules and generated output. (**B**) Table showing input data, complexity, computational resource and respective times required to perform the analysis

algorithms to produce NIM can be potentially very high. To overcome this problem, we reduced the problem of matching DNA interactions with nucleosome coordinates into a sorting algorithm. This in turns allows us to leverage the power of existing sorting algorithms in Python to find an optimized solution for generating NIM. The underlying algorithm in iNucs has a time complexity (n log n), where n is the larger of nucleosome counts and ligation junction counts reported in pairs file (Fig. 1B). One substantial benefit of reducing our problem into sorting is that iNucs can utilize the hardware-supported vectorization capabilities provided by Python libraries NumPy/Pandas that significantly speed up our program. Moreover, iNucs uses advanced parallelization techniques which exploits the power of CPU cores in the system. The detail for optimization of prepare module was explained in iNucs GitHub page.

Plot module: This module takes in the NIM from previous module and a query region of interest as input and produces an interactive heatmap plot showing the inter-nucleosome interaction counts within the query region. Here, we focused on the primary aim of the tool, i.e. generation of nucleosome-resolved interaction plots. An important feature of iNucs is that nucleosome-nucleosome interactions are divided in different classes based on ligation orientation of interaction pairs as this information is essential to study spatial arrangement of nucleosomes in the chromatin fiber (Supplementary Fig. S2A and B). Different color scales are used for ease of visualization and differentiation between different classes; purples (plus–minus, inward interactions), greens (minus–plus, outward interactions), reds and blues (plus–plus and minus–minus respectively, both classed as tandem interactions). As both classes of tandem interactions are indistinguishable topologically, they can additionally be represented together in grays. Further, all interactions could be summed up and visualized in oranges or overlaid as individual colors as per investigator's preference.

iNucs also benefits from an optional feature to compute normalized values for nucleosome-nucleosome interaction counts and plot the original and normalized matrices side by side (Supplementary Fig. S3). We have implemented the Observed/Expected non-zero normalization method described in hicTransform function from HiCExplorer tools (https://hicexplorer.readthedocs.io/en/latest/content/tools/hicTransform.html). The normalized value for each nucleosome pair (NP) is calculated as the observed interaction count for that NP divided by the expected count. The expected count for an NP is the average interaction count between nucleosome pairs whose distance is within the NP's average distance ± a given *distance*. The default value for the parameter *distance* is 200.

To evaluate the performance of iNucs in terms of computational time and memory usage, we applied iNucs on two sets of data for yeast and human (see evaluation and conclusion section), executed on a MacBook Pro (Fig. 1B).

3 Evaluation and conclusion

We used recently published *Saccharomyces cerevisiae* Hi-CO (Ohno *et al.*, 2019) and H1 human embryonic stem cell (hESC) line Micro-C (Krietenstein *et al.*, 2020) data to validate iNucs results. The genomic coordinates of nucleosomes for *S.cerevisiae* and ligation junctions (pairs file) were provided by the authors of Hi-CO data. In the case of human data, the pairs file was publicly available in 4DN portal, but there were no matched nucleosome coordinates. Therefore, we downloaded nucleosome coordinates from NucMap repository (https://ngdc.cncb.ac.cn/nucmap/NucMap_FTP_Directory/Homo_sapiens/byDataType/Nucleosome_peaks_DANPOS/) for H1-hESC (Sample ID hsNuc0070101) (Yazdi *et al.*, 2015; Zhao *et al.*, 2019). DANPOS (*Chen et al.*, 2013) was the main tool utilized by NucMap to call nucleosome peaks and coordinates from H1-hESC MNase-seq dataset. We plotted nucleosome-resolved interaction heatmap matrices for specific regions in the respective genomes and compared them with the related heatmaps published in original studies for those regions (Supplementary Fig. S2A and B). As nucleosome resolved heatmap was not available for H1-hESC, we focused on general chromatin organizational features. As expected, local chromatin features such as self-associated domains and boundaries were accurately reproduced by iNucs in the case of H1-hESC (Supplementary Fig. S2A). Also, iNucs generated nucleosome-resolved interaction matrices, which closely resembled nucleosome interaction profiles reported previously for yeast (Ohno *et al.*, 2019) (Supplementary Fig. S2B).

Acknowledgement

The authors thank Oliver Rando for helpful suggestions during the course of this study.

Funding

R.C.A. is a Wellcome Principal Research Fellow [095021, 200885]; the Wellcome Centre for Cell Biology is supported by core funding from Wellcome [203149]. This work was supported by grants-in-aid for Scientific Research (A) [20H00460], Japan Society for the Promotion of Science, Astellas Foundation for Research on Metabolic Disorders and Suntory Rising Stars Encouragement Program in Life Sciences (SunRiSE). M.O., N.S. and M.M.K. were supported with funding from Bristol Myers Squibb (BMS) company during this project.

Conflict of Interest: none declared.

References

Chen,K. *et al.* (2013) DANPOS: dynamic analysis of nucleosome position and occupancy by sequencing. *Genome Res.*, **23**, 341–351.

Kerpedjiev,P. *et al.* (2018) HiGlass: web-based visual exploration and analysis of genome interaction maps. *Genome Biol.*, **19**, 125.

Krietenstein,N. *et al.* (2020) Ultrastructural details of mammalian chromosome architecture. *Mol. Cell*, **78**, 554–565.e7.

Ohno,M. *et al.* (2019) Sub-nucleosomal genome structure reveals distinct nucleosome folding motifs. *Cell*, **176**, 520–534.e25.

Robinson,J.T. *et al.* (2018) Juicebox.js provides a cloud-based visualization system for Hi-C data. *Cell Syst.*, **6**, 256–258.e1.

Yazdi,PG. *et al.* (2015) Nucleosome organization in human embryonic stem cells. *PLoS One*, **10**, e0136314.

Zhao,Y. *et al.* (2019) NucMap: a database of genome-wide nucleosome positioning map across species. *Nucleic Acids Res.*, **47**, D163–D169.