COMPUTATIONAL
ANDSTRUCTURAL
BIOTECHNOLOGY
J O U R N A L

SnapHiC2: A computationally efficient loop caller for single cell Hi-C data

Xiaoqi Li [a,1], Lindsay Lee [b,1], Armen Abnousi [b], Miao Yu [c], Weifang Liu [d], Le Huang [e], Yun Li [d,f,g,*], Ming Hu [b,*]

[a] Carolina Health Informatics Program, University of North Carolina, Chapel Hill, NC, USA
[b] Department of Quantitative Health Sciences, Lerner Research Institute, Cleveland Clinic Foundation, Cleveland, OH, USA
[c] State Key Laboratory of Genetic Engineering, School of Life Sciences, Fudan University, Shanghai, China
[d] Department of Biostatistics, University of North Carolina, Chapel Hill, NC, USA
[e] Curriculum in Bioinformatics and Computational Biology, University of North Carolina, Chapel Hill, NC, USA
[f] Department of Genetics, University of North Carolina, Chapel Hill, NC, USA
[g] Department of Computer Science, University of North Carolina, Chapel Hill, NC, USA

A R T I C L E   I N F O

A B S T R A C T

Single cell Hi-C (scHi-C) technologies enable the study of chromatin spatial organization directly from complex tissues at single cell resolution. However, the identification of chromatin loops from single cells is challenging, largely due to the extremely sparse data. Our recently developed SnapHiC pipeline provides the first tool to map chromatin loops from scHi-C data, but it is computationally intensive. Here we introduce SnapHiC2, which adapts a sliding window approximation when imputing missing contacts in each single cell and reduces both memory usage and computational time by 70%. SnapHiC2 can identify 5 Kb resolution chromatin loops with high sensitivity and accuracy and help to suggest target genes for GWAS variants in a cell-type-specific manner. SnapHiC2 is freely available at: https://github.com/HuMingLab/SnapHiC/releases/tag/v0.2.2.

© 2022 The Author(s). Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY-NC-ND license (http://creative-commons.org/licenses/by-nc-nd/4.0/).

1. Introduction

The recently developed single cell Hi-C (scHi-C) [1–4] and its derived technologies, such as *sn*-m3c-seq [5,6], provide powerful tools to study chromatin spatial organization directly from complex tissues at single cell resolution. In particular, *sn*-m3c-seq can reveal distinct cell types based on single cell DNA methylome [5,6]. Many three-dimensional (3D) genome features, including A/B compartments, topologically associating domains (TADs) and chromatin loops, have been characterized in single cells [7–9]. Among them, it is of particular interest to identify chromatin loops from single cells, which can shed novel insights on cell-type-specific gene regulation mechanisms.

Chromatin loops, which are defined as pairs of genomic loci with significantly higher contact frequency compared to the expected contact frequency due to random collision, have been originally discovered from deeply sequenced bulk cell Hi-C data [10] and play a critical role in genome structure and genome

function [11]. However, identifying chromatin loops from single cell Hi-C (scHi-C) data is extremely challenging, largely due to data sparsity. Specifically, ~1 billion contacts are usually required to robustly map chromatin loops from bulk cell Hi-C data. Nevertheless, most currently available scHi-C and *sn*-m3c-seq datasets consist of a few hundred cells for each cell type, with on average ∼ 1 million contacts per cell [2,5,6]. Aggregating all cells belonging to the same cell type into pseudo bulk Hi-C data still cannot reach the typical required number of contacts for loop calling, limiting the sensitivity of existing methods developed for bulk Hi-C data [12]. A detailed description of publicly available scHi-C datasets [1–6,13–20] and related computational challenges can be found in recent review papers[21–23].

To address this challenge, we recently published SnapHiC, a computational pipeline to identify loops from scHi-C data [9]. SnapHiC consists of two steps. First, it adopts the random walk with restart (RWR) algorithm [7] to impute contact probability in each cell at 10 Kb bin resolution. Next, it applies both global and local background models to identify chromatin loops. The key methodological innovation in SnapHiC is that each cell is treated as an independent unit, leading to substantially improved power in loop calling (see details in our recent review paper [21]).

---

* Corresponding authors.
  E-mail addresses: yun_li@med.unc.edu (Y. Li), hum@ccf.org (M. Hu).
  [1] Co-first authors.

However, the computational cost of SnapHiC is high. In particular, the first step (i.e., RWR imputation) consumes a large amount of computational resource in terms of both computing time and required memory, thus limiting its applicability to a large number (e.g., greater than500) of cells or calling loops at sub-10 Kb resolution. Given the increasing interest in identifying loops from scHi-C data [24], a computationally more efficient implementation of SnapHiC is therefore of urgent need.

Here we present SnapHiC2, an efficient chromatin loop caller for scHi-C data. Compared to SnapHiC, SnapHiC2 achieves more than 3X speed up and saves $\sim$ 70% memory usage, making the identification of sub-10 Kb-resolution loops from a large number of cells computationally feasible. SnapHiC2 is freely available at: https://github.com/HuMingLab/SnapHiC/releases/tag/v0.2.2.

2. Materials and methods

2.1. A sliding window approach for scHi-C imputation.

In SnapHiC2, we employ a sliding window approach to approximate the RWR algorithm (**Figure S1**). Briefly, in each single cell, we model each chromosome as an unweighted undirected graph, where each node is an equal-sized genomic locus (e.g., 5 Kb bin or 10 Kb bin). We then add edges to adjacent genomic loci pairs and all genomic loci pairs with chromatin contacts. Next, we use the RWR algorithm to calculate the probability of traveling between any two loci, which is defined as the imputed chromatin contact probability. The computational complexity of the RWR algorithm is determined by the size of the graph, in other words, the number of genomic loci in each chromosome. In the original study [9], we applied the RWR algorithm to the entire chromosome, which is time-consuming and requires a large amount of memory. As a computationally more efficient implementation, SnapHiC2 leverages partially overlapped sliding window approximation, and substantially speeds up the RWR algorithm.

Let $X = \{x_{mn}\}$ represent a 10 Kb bin resolution Hi-C contact matrix for one cell (the bin resolution can also be set to 5 Kb), where x_{mn} is the number of contacts between bin m and bin n. Instead of computing the RWR-imputed contact probability for all 10 Kb bin pairs in the entire chromosome-wide Hi-C contact matrix, we break down the computation into a series of partially overlapped matrices along the diagonal line of the original matrix. Specifically, each matrix is d by d in size, and two consecutive matrices overlap by a $d/2$ by $d/2$ sub-matrix. In other words, the step size of the sliding window is $d/2$ (see detailed illustration in **Figure S1**).

Let $A_k = \{a_{ij}^k : 1 \leq i < j \leq d\}$ be a d by d matrix, which is the k th step in the moving window. We use the RWR algorithm to impute the contact probability for all 10 Kb bin pairs within this d by d matrix. To avoid artifacts near corners due to incomplete chromatin contact information, we only keep imputed contact probability for a subset of bin pairs in the middle rectangle specified below:

$$\left\{ a_{ij}^k : 1 \leq i < j \leq d, j - i \leq d/2, i + j \geq d/2 + 1, i + j \leq 3d/2 \right\}.$$

Figure S1 illustrates the first 3 steps of the sliding window procedure. In step 1, we perform RWR for bin pairs in the red block and generate imputed contact probabilities for bin pairs in the red rectangle. Then moving to step 2, we again perform RWR for bin pairs in the blue block and generate imputed contact probabilities for bin pairs in the blue rectangle. Continuing to step 3, we yet again perform RWR for bin pairs in the purple block and generate imputed contact probabilities for bin pairs in the purple rectangle. We repeat this sliding window procedure until we reach the end of

the chromosome. Then, we concatenate the RWR-imputed contact probabilities for all bin pairs within in 1 Mb genomic distance (the 1 Mb boundary is highlighted by the dashed yellow line, and bin pairs subject to loop calling reside in shaded regions illustrated by the striped red, blue, purple lines), while skipping the two corners near chromosome start and end (i.e., the black triangles).

2.2. A data-driven approach to determine sliding window size

It is critical to choose an appropriate size for the sliding windows. When the window size is too small, a large amount of information outside the sliding window will be lost, resulting in an inaccurate approximation of RWR-imputed contact probabilities. In contrast, when the window size is too large, the computational cost will be too expensive. SnapHiC2 adopts a data-driven approach and selects the window size to keep more than 80% of the chromatin contacts, achieving a balance between imputation accuracy and computational efficiency (see details in **Results**).

3. Results

3.1. The data-driven approach to determine sliding window size

We used scHi-C data from 100 oligodendrocytes (ODCs) from human cortex [5] and 742 mouse embryonic stem cells (mESCs) [2] as benchmark examples. For each cell, we calculated the proportion of intra-chromosomal contacts within a certain 1D genomic distance (**Figure S2A**) and found that more than 80% of contacts were within 20 Mb and 10 Mb, for 100 ODCs and 742 mESCs, respectively.

Next, we applied SnapHiC2 with five different sliding window sizes (10 Mb, 20 Mb, 30 Mb, 40 Mb, and 50 Mb) and compared SnapHiC2-identified chromatin loops with SnapHiC-identified chromatin loops, which are based on the full chromosome version of the RWR imputation. We first evaluated the loop overlap between these two versions. SnapHiC2 identified a slightly larger number of loops than SnapHiC (**Figure S2B**), with 60.9% ~77.3% of loops identified as overlapping (**Figure S2C**). We then evaluated the precision, recall, and F1-score (i.e., the harmonic mean of precision and recall), treating as working truth chromatin loops identified from bulk Hi-C data, H3K4me3 PLAC-seq data, as well as cohesin and H3K27ac HiChIP data [25–29]. We found that SnapHiC2 with different sliding window sizes achieved comparable precision, recall, and F1-score, compared with SnapHiC (**Figure S2D, S2E, and S2F**). For example, when applied to 100 ODCs, SnapHiC2 with sliding window size 20 Mb achieved precision 0.699, recall 0.261 and F1-score 0.380, similar to SnapHiC (precision 0.775, recall 0.232 and F1-score 0.357). Consistently, when applied to 742 mESCs, SnapHiC2 with sliding window size 10 Mb achieved precision 0.560, recall 0.285 and F1-score 0.378, again comparable to SnapHiC (precision 0.655, recall 0.278 and F1-score 0.391). Taken together, our results suggest that SnapHiC2, with its data-driven strategy to select sliding window size that retains more than 80% of contacts, can identify loops with similar quality as the original SnapHiC algorithm.

3.2. SnapHiC2 is computationally efficient

Reassured with the satisfactory performance of SnapHiC2, we then evaluated its computational efficiency. Specifically, for the RWR imputation step, we applied both SnapHiC and SnapHiC2 (with sliding window size 20 Mb) to 100 ODCs and 742 mESCs, each repeated three times, and recorded the corresponding computational time and memory used (Fig. 1). When applied to 100 ODCs, SnapHiC2 was 3.2 times faster than SnapHiC (average time: 1.56 h
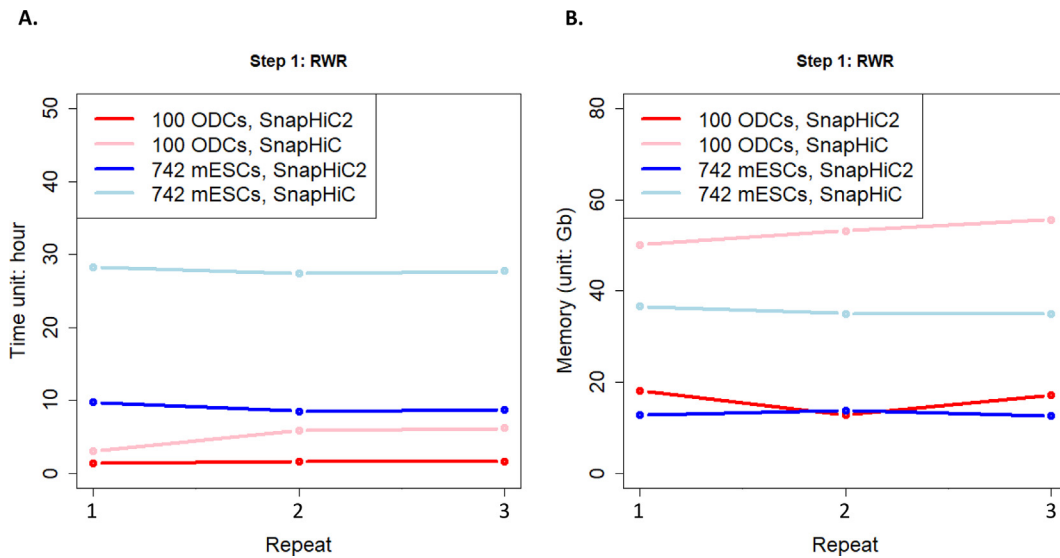
Figure 1. Computational advantage of SnapHiC2 in speed and memory, using 100 ODCs and 742 mESCs as the illustrative example. A. Computational time comparison. B. Memory usage comparison.

vs. 5.04 h, Fig. 1A) and only consumed 30.2% of the memory (average required memory: 16.02 GB vs. 53.02 GB, Fig. 1B). Consistently, when applied to 742 mESCs, SnapHiC2 was 3.1 times faster than SnapHiC (average time: 8.98 h vs. 27.81 h, Fig. 1A), and only consumed 36.7% of the memory (average required memory: 13.04 GB vs. 35.55 GB, Fig. 1B). These results demonstrate that SnapHiC2 substantially enhances computational efficiency compared to SnapHiC.

3.3. SnapHiC2 can identify 5 Kb resolution chromatin loops with high sensitivity and accuracy

In the original study [9], we applied SnapHiC to identify 10 Kb resolution chromatin loops. With the more efficient SnapHiC2 implementation, we can afford to improve the bin resolution to 5 Kb. Here, we re-analyzed scHi-C data in chromosome 3 from 742 mESCs [2] at 5 Kb resolution using both SnapHiC and SnapHiC2. In addition, we aggregated the 742 mESCs and applied two versions of the HiCCUPS algorithm [10], one with the default settings (HiCCUPS-default) for deeply sequenced bulk Hi-C data (greater than1 billion contacts) and the other with more lenient settings (HiCCUPS-lenient) for shallowly sequenced bulk Hi-C data (~500 million contacts).

We first applied SnapHiC, SnapHiC2, HiCCUPS-default, and HiCCUPS-lenient to chromosome 3 data in all 742 mESCs, and a subset of mESCs (N = 100, 200, 300, 400, 500, 600, and 700, Fig. 2). We randomly sampled a subset of mESCs three times to evaluate the robustness of SnapHiC2, HiCCUPS-default and HiCCUPS-lenient. Due to the high computational cost of SnapHiC at 5 Kb bin resolution, we only run SnapHiC once. We found that both SnapHiC and SnapHiC2 identified more 5 Kb loops than both two versions of HiCCUPS (Fig. 2A and **Table S1**). For example, using all 742 mESCs, SnapHiC, SnapHiC2, HiCCUPS-default and HiCCUPS-lenient identified 252, 294, 6 and 24 5 Kb loops in chromosome 3, respectively (Fig. 2A). Next, we used 5 Kb resolution loops identified from bulk Hi-C data, H3K4me3 PLAC-seq and cohesin, and H3K27ac HiChIP data [26–29] as the reference loop list (i.e., treated as the working truth) and evaluated the precision, recall, and F1-score. Compared to loops identified from two versions of HiCCUPS, both SnapHiC and SnapHiC2-identified loops achieved slightly lower precision (Fig. 2B), much improved recall (Fig. 2C), and substantially higher F1-score (Fig. 2D). For example, using all 742

mESCs, SnapHiC2 loops have precision 0.534, recall 0.182 and F1-score 0.271, while SnapHiC loops have precision 0.623, recall 0.180 and F1-score 0.279. In addition, HiCCUPS-default loops have precision 0.667, recall 0.006 and F1-score 0.013. HiCCUPS-lenient loops have precision 0.708, recall 0.031 and F1-score 0.060.

3.4. The Sox2 locus

We then examined the *Sox2* gene locus, where a functional enhancer-promoter interaction between *Sox2* promoter and the super-enhancer ~ 100 Kb downstream has previously been validated by CRISPR experiment in mESCs [30]. At 5 Kb resolution, both SnapHiC and SnapHiC2 required as few as 100 cells to identify such enhancer-promoter interaction, whereas HiCCUPS-default and HiCCUPS-lenient required at least 600 cells and 500 cells, respectively (Fig. 2E). This example showcases the significantly enhanced sensitivity of SnapHiC2 compared to two versions of HiCCUPS.

Next, for all 742 mESCs, we compared the SnapHiC2-identified 5 Kb-resolution loop with the SnapHiC-identified 10 Kb-resolution loop with *Sox2* promoter (Fig. 2F). The bin interacting with *Sox2* promoter is chr3:34.755 Mb-34.76 Mb and chr3:34.75 Mb-34.76 Mb, based on SnapHiC2′s 5 Kb calling and SnapHiC's 10 Kb calling, respectively. In the CRISPR experiment [30], the deleted region is chr3:34,756,113–34,761,855, largely overlapped with the 5 Kb interacting bin identified by SnapHiC2. In addition, we found a strong H3K27ac ChIP-seq peak (chr3:34,754,670–34,7 58,019) from mESCs overlapped with the 5 Kb interacting bin. Taken together, these pieces of evidence suggest that the 5 Kb loops identified by SnapHiC2 can better pinpoint the functional enhancer of *Sox2* gene than the 10 Kb loops identified by SnapHiC.

3.5. SnapHiC2 can identify cell-type-specific loops linking GWAS variants to putative disease genes.

Lastly, we applied SnapHiC2 to identify 5 Kb-resolution loops from astrocytes, L2/3 neurons, microglia and oligodendrocytes, each with 261 cells [5] (**Table S2**). We identified a large proportion of cell-type-specific loops, underscoring the importance of mapping chromatin loops using single cells from the same cell type (**Figure S3**). As one illustrative example, we checked chromatin loops anchored at the transcription start site of gene *PAGR1*, which
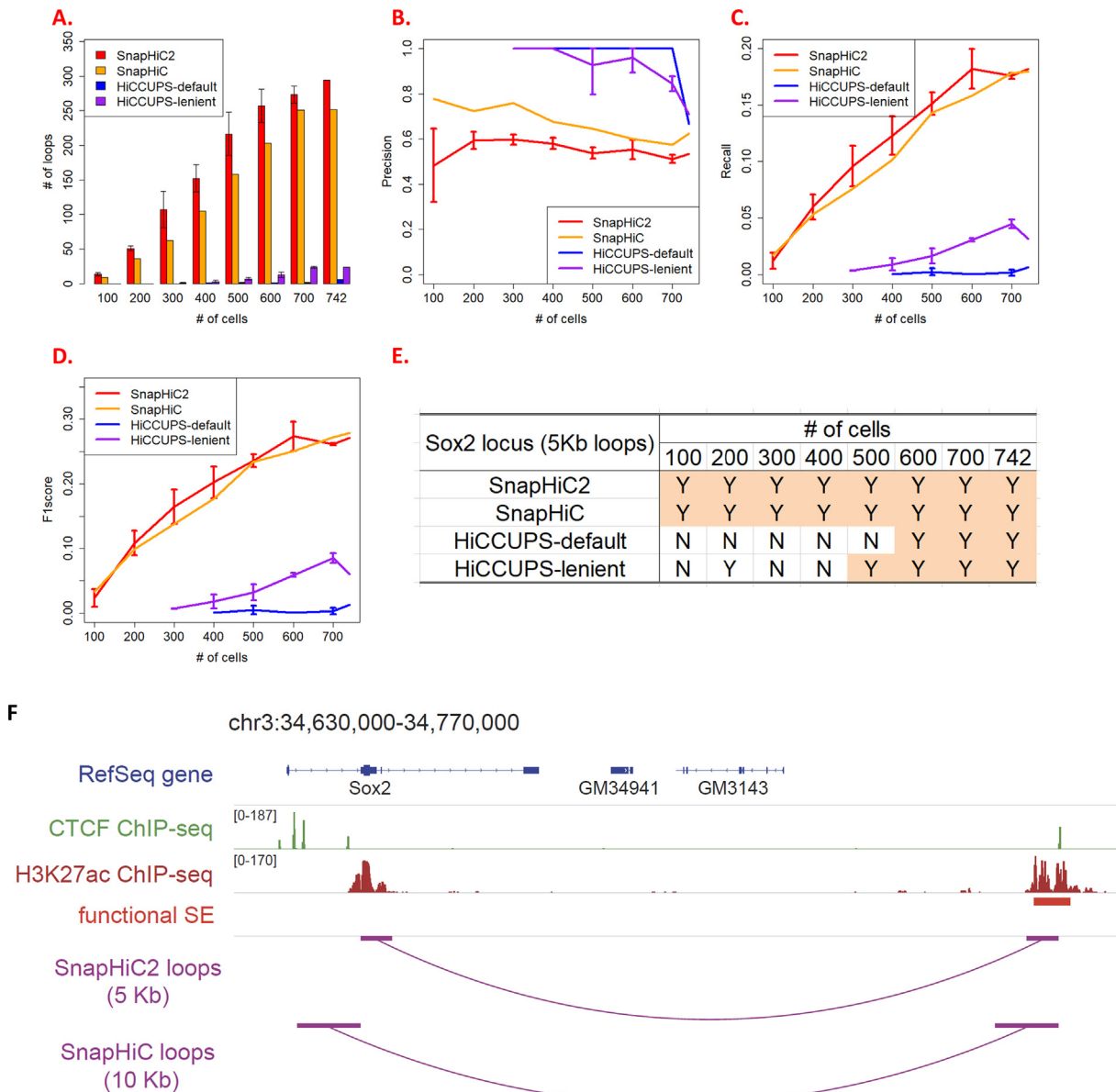
Figure 2. SnapHiC2 can identify high-quality 5Kb resolution chromatin loops from mESC cells. In Figure 2A, 2B, 2C and 2D, the error bar represents the standard deviation across three random samples. A. For 100, 200, 300, 400, 500, 600, 700, 742 cells, SnapHiC2 identifies similar numbers of loops with SnapHiC. SnapHiC2 identified more loops than HiCCUPS with either default or lenient settings. B. SnapHiC2 achieves slightly lower precision than SnapHiC and the two versions of HiCCUPS. C. SnapHiC2 achieves slightly higher recall than SnapHiC and much higher recall than the two versions of HiCCUPS. D. SnapHiC2 achieves comparable F1-score with SnapHiC and much higher F1-score than the two versions of HiCCUPS. E. Both SnapHiC and SnapHiC2 detect a functional 5Kb resolution enhancer-promoter interaction at the *Sox2* locus in mESCs with much fewer cells than the two versions of HiCCUPS. F. Genome browser view of the *Sox2* locus showing that the 5Kb loops identified by SnapHiC2 can better pinpoint the functional enhancer of *Sox2* gene than the 10Kb loops identified by SnapHiC.

has been reported to be associated with neuropsychiatric disorders [31]. We found a L2/3-specific loop linking the promoter of gene *PAGR1* to a 5 Kb bin (chr16:29,930,000–29,935,000) containing six schizophrenia-associated GWAS SNPs (Fig. 3) (rs4407079 [chr16:29,931,899], rs4609871 [chr16:29,932,064], rs9936474 [chr16:29,932,691], rs12932403 [chr16:29,934,050], rs12596042 [chr16:29,934,754] and rs11150574 [chr16:29,934,819]). Among them, four SNPs (rs4407079, rs4609871, rs9936474 and rs12932403) reside in neuronal enhancers [25]. In addition, gene *PAGR1* shows the highest expression in L2/3 neurons among these four cell types [32] (Fig. 3). Notably, gene *PARG1* is greater than 100 Kb away from these schizophrenia-associated GWAS SNPs, with multiple genes residing in the middle. In addition, multiple schizophrenia-associated GWAS SNPs are outside of the neu-

ronal enhancers. Integrating the multi-omics evidence from epigenome, transcriptome and 3D genome suggests that *PAGR1* is the likely functional gene underlying the GWAS variants, and most likely in L2/3 neurons.

4. Discussion

In this paper, we describe SnapHiC2, a computationally efficient pipeline to identify chromatin loops from scHi-C data. Compared to our original SnapHiC, SnapHiC2 adopts a sliding window approach when implementing the RWR algorithm, achieving more than 3 times speed up and reducing memory usage by around 70%. In addition, we provide an implementation of the loop calling step
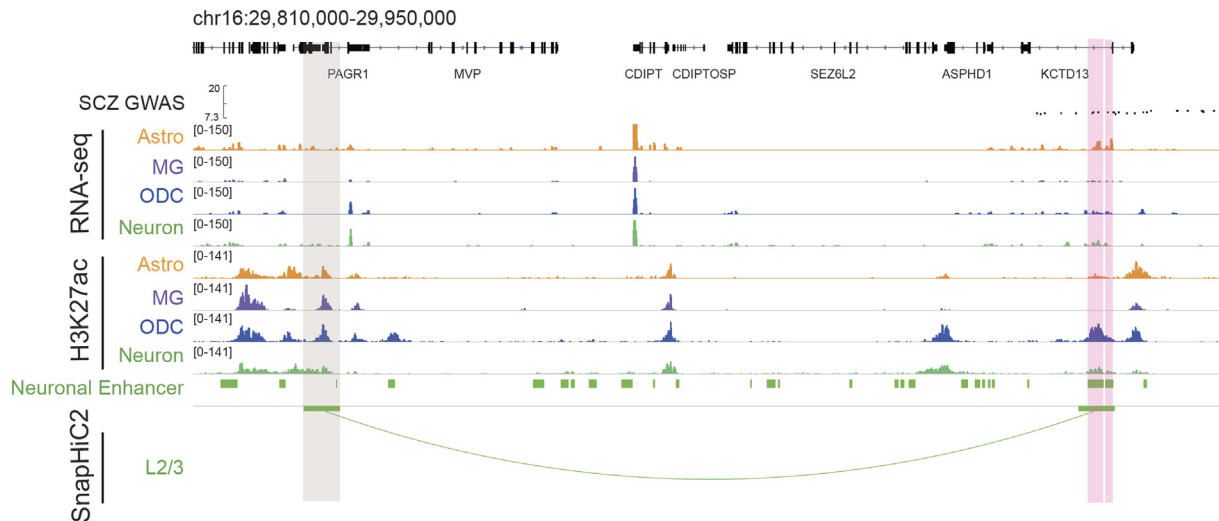
Figure 3. SnapHiC2 identifies a 5Kb L2/3-specific loop linking the *PAGR1* promoter to six schizophrenia-associated GWAS SNPs.

(i.e., step 2) for running on GPUs, leveraging the JAX library in Python. When testing on the Volta GPU with Intel(R) Xeon(R) Gold 6148 CPU @ 2.40 GHz, we found that SnapHiC2's GPU implementation of step 2 achieved more than 2X speed-up compared with SnapHiC's CPU implementation of step 2, each with a single processor. SnapHiC2 enables the identification of chromatin loops from single cells at 5 Kb resolution, facilitating the characterization of promoter-interacting enhancers, as well as identification and prioritization of putative target genes for GWAS variants. We believe that SnapHiC2 will become a useful tool to study 3D chromatin organization at single cell resolution.

Funding

Reference genomes

We used mm10 and hg19 for mouse scHi-C data and human scHi-C data, respectively.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. A.A. is currently an employee of RocketReach.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.csbj.2022.05.046.

References

[1] Nagano T et al. Single-cell Hi-C reveals cell-to-cell variability in chromosome structure. Nature 2013;502:59–64. https://doi.org/10.1038/nature12593.

[2] Nagano T et al. Cell-cycle dynamics of chromosomal organization at single-cell resolution. Nature 2017;547:61–7. https://doi.org/10.1038/nature23001.

[3] Flyamer IM et al. Single-nucleus Hi-C reveals unique chromatin reorganization at oocyte-to-zygote transition. Nature 2017;544:110–4. https://doi.org/10.1038/nature21711.

[4] Stevens TJ et al. 3D structures of individual mammalian genomes studied by single-cell Hi-C. Nature 2017;544:59–64. https://doi.org/10.1038/nature21429.

[5] Lee DS et al. Simultaneous profiling of 3D genome structure and DNA methylation in single human cells. Nat Methods 2019;16:999–1006. https://doi.org/10.1038/s41592-019-0547-z.

[6] Liu H et al. DNA methylation atlas of the mouse brain at single-cell resolution. Nature 2021;598:120–8. https://doi.org/10.1038/s41586-020-03182-8.

[7] Zhou J et al. Robust single-cell Hi-C clustering by convolution- and random-walk-based imputation. PNAS 2019;116:14011–8. https://doi.org/10.1073/pnas.1901423116.

[8] Zhang R, Zhou T, Ma J. Multiscale and integrative single-cell Hi-C analysis with Higashi. Nat Biotechnol 2021. https://doi.org/10.1038/s41587-021-01034-y.

[9] Yu M et al. SnapHiC: a computational pipeline to identify chromatin loops from single-cell Hi-C data. Nat Methods 2021;18:1056–9. https://doi.org/10.1038/s41592-021-01231-2.

[10] Rao SSP et al. A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. Cell 2014;159:1665–80.

[11] Grubert F et al. Landscape of cohesin-mediated chromatin loops in the human genome. Nature 2020;583:737–43. https://doi.org/10.1038/s41586-020-2151-x.

[12] Li X, An Z, Zhang Z. Comparison of computational methods for 3D genome analysis at single-cell Hi-C level. Methods 2019. https://doi.org/10.1016/j.ymeth.2019.08.005.

[13] Ramani V et al. Massively multiplex single-cell Hi-C. Nat Methods 2017;14:263–6. https://doi.org/10.1038/nmeth.4155.

[14] Tan L et al. Changes in genome architecture and transcriptional dynamics progress independently of sensory experience during post-natal brain development. Cell 2021;184:741–758.e717. https://doi.org/10.1016/j.cell.2020.12.032.

[15] Tan L, Xing D, Chang CH, Li H, Xie XS. Three-dimensional genome structures of single diploid human cells. Science (New York 2018;N.Y.) 361:924–8. https://doi.org/10.1126/science.aat5641.

[16] Tan L, Xing D, Daley N, Xie XS. Three-dimensional genome structures of single sensory neurons in mouse visual and olfactory systems. Nat Struct Mol Biol 2019;26:297–307. https://doi.org/10.1038/s41594-019-0205-2.

[17] Li G et al. Joint profiling of DNA methylation and chromatin architecture in single cells. Nat Methods 2019;16:991–3. https://doi.org/10.1038/s41592-019-0502-z.

[18] Collombet S et al. Parental-to-embryo switch of chromosome organization in early embryogenesis. Nature 2020;580:142–6. https://doi.org/10.1038/s41586-020-2125-z.

[19] Kim HJ et al. Capturing cell type-specific chromatin compartment patterns by applying topic modeling to single-cell Hi-C data. PLoS Comput Biol 2020;16: e1008173.

[20] Mulqueen RM et al. High-content single-cell combinatorial indexing 2011. 2021,;2021(2001):. https://doi.org/10.1101/2021.01.11.425995. %J bioRxiv 425995.

[21] Yu M, Li Y, Hu M. Mapping chromatin loops in single cells. Trends in genetics : TIG 2022. https://doi.org/10.1016/j.tig.2022.03.007.

[22] Zhou T, Zhang R, Ma J. The 3D Genome Structure of Single Cells. Annual review of biomedical data science 2021;4:21–41. https://doi.org/10.1146/annurev-biodatasci-020121-084709.

[23] Galitsyna AA, Gelfand MS. Single-cell Hi-C data analysis: safety in numbers. Briefings Bioinf 2021;22. https://doi.org/10.1093/bib/bbab316.

[24] Chen S, Wang J, Gao X, Li Y-L. A general and robust deep learning framework for accurate chromatin loop detection in genome-wide contact maps. 202220012030478367 2022. https://doi.org/10.1101/2022.01.30.478367. %J bioRxiv.

[25] Nott A et al. Brain cell type-specific enhancer-promoter interactome maps and disease-risk association. Science (New York, NY) 2019;366:1134–9. https://doi.org/10.1126/science.aay0793.

[26] Bonev B et al. Multiscale 3D Genome Rewiring during Mouse Neural Development. Cell 2017;171:557–572.e524. https://doi.org/10.1016/j.cell.2017.09.043.

[27] Juric I et al. MAPS: model-based analysis of long-range chromatin interactions from PLAC-seq and HiChIP experiments. *PLoS Comput Biol* **In press** 2019. https://doi.org/10.1101/411835.

[28] Mumbach MR et al. HiChIP: efficient and sensitive analysis of protein-directed genome architecture. Nat Methods 2016;13:919–22. https://doi.org/10.1038/nmeth.3999.

[29] Mumbach MR et al. Enhancer connectome in primary human cells identifies target genes of disease-associated DNA elements. Nat Genet 2017;49:1602–12. https://doi.org/10.1038/ng.3963.

[30] Li Y et al. CRISPR reveals a distal super-enhancer required for Sox2 expression in mouse embryonic stem cells. PLoS ONE 2014;9:e114485.

[31] Birnbaum R, Jaffe AE, Hyde TM, Kleinman JE, Weinberger DR. Prenatal expression patterns of genes associated with neuropsychiatric disorders. The American journal of psychiatry 2014;171:758–67. https://doi.org/10.1176/appi.ajp.2014.13111452.

[32] Zhang Y et al. Purification and Characterization of Progenitor and Mature Human Astrocytes Reveals Transcriptional and Functional Differences with Mouse. Neuron 2016;89:37–53. https://doi.org/10.1016/j.neuron.2015.11.013.