Chromosight: A computer vision program for pattern detection in chromosome contact maps

Cyril Matthey-Doret[1,2], Lyam Baudry[1,2], Axel Breuer[1,3], Rémi Montagne[1], Nadège Guiglielmoni[1,4], Vittore Scolari[1], Etienne Jean[1], Arnaud Campeas[3], Philippe-Henri Chanut[3], Edgar Oriol[3], Adrien Meot[3], Laurent Politis[3], Antoine Vigouroux[5], Pierrick Moreau[1], Romain Koszul[1] and Axel Cournac[1]

[1] Institut Pasteur, Unité Régulation Spatiale des Génomes, UMR 3525, CNRS, Paris, F-75015, France
[2] Sorbonne Université, Collège Doctoral, F-75005, Paris, France
[3] ENGIE Global Energy Management, 92930, Paris, France
[4] Service Evolution Biologique et Ecologie, Avenue F.D. Roosevelt 50, 1050, Brussels, Belgium
[5] Synthetic Biology Laboratory, Institut Pasteur, 75015, Paris, France

Corresponding author: rkoszul@pasteur.fr,acournac@pasteur.fr

Abstract *Chromosomes of all species studied so far display a variety of higher-order organizational features such as domains, loops, or compartments. Many of these structures have been characterized from the genome-wide contact maps generated by chromosome conformation capture approaches (Hi-C, ChIA-PET,...). Indeed, DNA 3D structures translate as distinct patterns visible on these maps. We developed* Chromosight*, an algorithm based on computer vision approaches that automatically detect and quantify any type of pattern in contact data. Chromosight detects 3 times as many patterns as existing programs, while being faster and fit to any genome, including small, compact ones. Chromosight is user-friendly and can be extended to user-provided patterns. We validated the program by applying it to a variety of chromosomal structures found in mammals. Code and documentation: https://github.com/koszullab/chromosight*

Keywords Domain borders, Genomics, Loops, Hi-C, Detection, computer vision

Introduction

Proximity ligation derivatives of the chromosome conformation capture (3C) approach [1] such as Hi-C [2] have unveiled a wide variety of chromatin 3D arrangements of potential interest regarding chromosome metabolic processes. Indeed, these approaches reveal the average contact frequencies between DNA segments within a genome, computed over hundreds of thousands of cells. These frequencies reflect the relative spatial distances separating these regions. In all species studied so far, chromosomes are sub-divided into sub-Mb domains. In mammals, topologically associating domains (TADs) are relatively stable self-interacting regions formed and maintained by the action of the structural maintenance of chromosomes (SMC) protein complex cohesin [3,4,5,6]. TADs have been proposed to emerge from a loop extrusion mechanism, in which cohesins would enlarge DNA loops between two roadblocks along the chromosomes. These roadblocks are formed by the CCCTC-binding factor (CTCF) [7]., enriched at TADs borders. A number of experimental and computational studies suggest that TADs may serve as scaffolds for gene regulation [8] Chromatin loops connecting distant loci across the genome (from a few kb to several Mb) are also common features of chromosome architecture and have been detected by Hi-C along yeast metaphase [9] and mammalian interphase chromosomes [10]. In mammals, these loops frequently bridge CTCF-binding sites, at the extremities of TADs, and are dependent on cohesin. The regulation of cohesin-dependent loops appears to be conserved from yeast to mammals, suggesting a ubiquitous mechanism that evolved to promote different functions [9].

Most structural features can be identified by eye on a Hi-C contact map. This identification is sometimes carried out manually, which may prove unwieldy or impractical for large or noisy datasets. Several methods have been developed to identify specifically looping interactions in Hi-C contact maps. For instance, diffHiC [11] looks for contact enrichments between pairs of loci. Other tools explicitly look for loop patterns using tailored rules: HiCExplorer [12] computes statistical distributions from the Hi-C contacts and looks for groups of outlier pixels forming neighbourhoods, while tools like HOMER [13] or HiCCUPS [10] compare the intensity of each pixel with a surrounding region. However, most of

these tools remain perfectible. First, they were developed to investigate loops in humans, *e.g.* discrete dots positioned at relatively large distances from the main diagonal of Hi-C contact maps, and very different from loops found so far in the smaller, more compact genomes of bacteria and fungi. Second, they miss many loops otherwise clearly visible by eye and hence suffer from a low detection rate. Most of these tools are also computationally intensive and require either a dedicated GPU (HiCCUPS) or a long run-time (*e.g.* HOMER). Some groups have recently started to use kernel convolutions to tackle the latter limitation. Notably, the cooltools suite (https://github.com/mirnylab/cooltools) has implemented a "dot finder" algorithm that uses the same surrounding regions method as HiCCUPS, but uses kernel convolution instead of explicit comparisons to speed up operations.

Here we introduce *Chromosight*, a program that automatically detects generic patterns in chromosome contact maps, with a specific focus on chromosomal loops and domain borders. Chromosight is a user-friendly python package, with minimal installation requirements. It can be applied to any contact map, independently of species, protocol or genome size. The source code is available on github `https://www.github.com/koszullab/chromosight` and our implementation is readily available on PyPi and bioconda as a standalone package. We benchmarked its precision and recall rate on simulated datasets and compared it to existing algorithms, showing it outperforms all other available programs. It is also markedly faster than most of these. Importantly, it works well on any genome and any pattern. The approach can easily be extended to user-defined structures visible on a contact map, such as cohesin injection points, or centromere clustering.

Results

Presentation of the algorithm

Chromosight takes a single, whole-genome contact map as an input in `cool` format [14], and starts by pre-processing each chromosome's submatrix to enhance local variations in the signal (Fig. 1a, methods). Intra-chromosomal contacts above a user-defined distance are discarded to constrain the analysis to relevant scales and improve performance. The core task of the Chromosight's algorithm consists in detecting a given template (e.g. loop or border kernel) within an image (i.e. the Hi-C matrix). This task is known as template matching and has been commonplace for a long time in the computer vision community [15]. Like most template matching procedures, Chromosight proceeds in 2 steps: 1) a correlation step where each sub-image is correlated to the template and 2) a selection step where sub-images with highest correlation values are labelled as template representations.

Convolution algorithms are often used in computer vision where images are typically dense. Hi-C contact maps, on the other hand, are extremely sparse. Chromosight's convolution algorithm is therefore designed to be fast and memory efficient on sparse matrices. For selecting contiguous regions of high correlation values, Chromosight uses connected component labelling (CCL). By converting the thresholded correlation map into a sparse adjacency graph, Chromosight can take advantage of an existing CCL implementation optimized for graphs to minimize both running time and memory usage.

Unlike other tools which rely on tailored scoring methods for each pattern type (*e.g.* Arrowhead for TAD detection [10]), Chromosight uses a single algorithm to detect built-in (loops, borders, hairpins...) or user-defined patterns. Regardless of the pattern, each detected instance is associated with a score (Pearson correlation), facilitating the interpretation of the result.

We assessed the performances of Chromosight for loop detection by benchmarking it against 4 existing programs on synthetic Hi-C data (Methods). We found that Chromosight has comparable precision (proportion of false positive calls) to state-of-the-art algorithms while having much higher recall rates (higher proportion of true positives detected) (Fig. 1b). Moreover, Chromosight's speed is comparable to the fastest tools available. For instance, on a machine with a 12 threads Intel i7-8700k CPU at 3.7GHz, Chromosight took 4 minutes and 4.4GB of RAM to perform loop detection at up to 20Mbp interaction distance on a human matrix with 249M contacts at a resolution of 10kb [16].

Chromosight can run either in the aforementioned detection mode, or in quantification mode (*quantify*). Chromosight *quantify* takes an input set of coordinates and returns their Pearson coefficients with a desired kernel. We used *quantify* to precisely measure the spatial scales at which cohesin loops

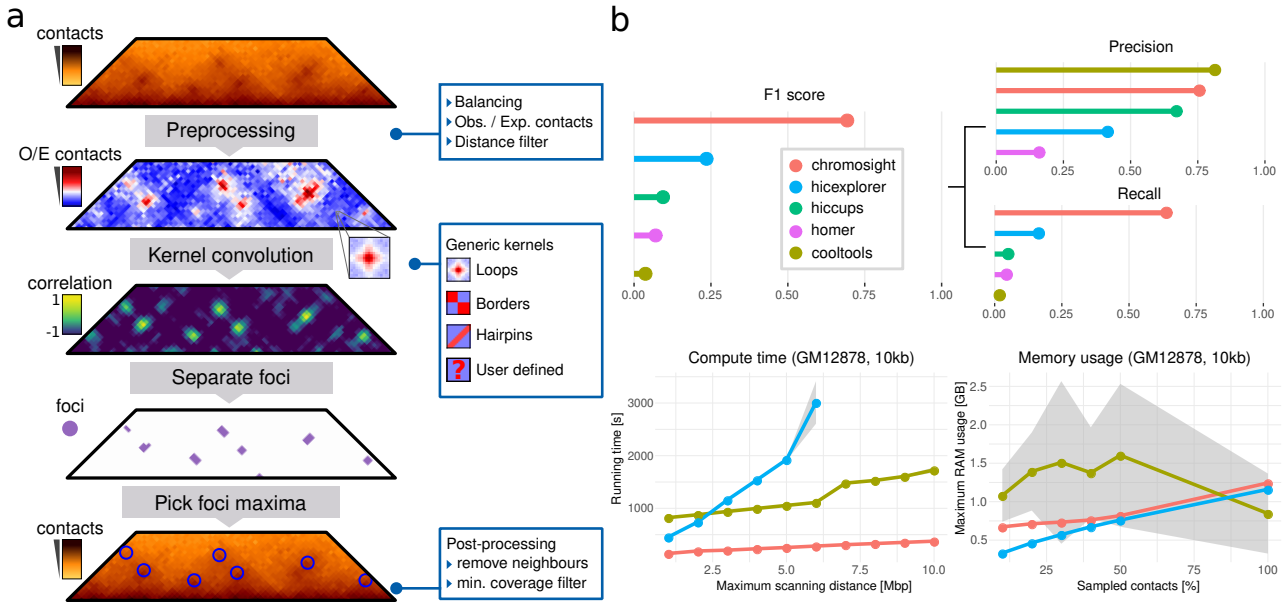

Fig. 1. Chromosight algorithm workflow and benchmark. a, Matrix preprocessing involves normalization balancing followed by the computation of observed / expected contacts. Only contacts between bins separated by a user-defined maximum distance are considered. The preprocessed matrix is then convolved with a kernel representing the pattern of interest. For each pixel of the matrix, a Pearson correlation coefficient is computed between the kernel and the surrounding window. A threshold is applied on the coefficients and a connected component labelling algorithm is used to separate groups of pixels (*i.e.* foci) with high correlation values (Methods). For each focus, the coordinate with the highest correlation value is used as the pattern coordinate. Coordinates located in poorly covered regions are discarded (Methods). **b,** Comparison of Chromosight with different loop callers. Top: F1 score, Precision and Recall score assessed on labelled synthetic Hi-C data (Methods). Higher is better. Bottom: performance of the different algorithms. Run-time and memory usage according to maximum scanning distance and the amount of downsampled contact events, respectively. The performance benchmark was run 5 times on data from human lymphoblastoid cell line (GM12878) Hi-C maps [16]. Means and standard deviations (grey areas) are plotted.

act. This allows us to compute what a loop "spectra", *i.e.* loop scores for pairs of cohesin peaks separated by increasing genomic distances (see Methods).

Detection and quantification in mammals

We applied Chromosight on published human and mouse Hi-C data generated in different laboratories. First, we searched for loops, borders, and hairpins in genome-wide contact maps generated from lymphoblastoids (GM12878) [16] (**Fig 2a**).

Loop detection yielded more than 18,000 occurrences (**Fig 2b**). The majority of the loop basis ($\simeq$55%, $p < 10^{-16}$, Fisher test)) fall into loci enriched in cohesin subunit Rad21, as expected[17]. Multiple loops often originate from a same basis, reflecting either an heterogeneity of structures in the population, the formation of rosette-like structures, or both.

We then compared loop scores between wild type (WT) and mutant Hela cells where cohesin was depleted [17]. Loop detection in WT Hela cells yields similar results as with GM12878 cells, with 15,600 loops. The loop scores of these WT borders in cohesin-depleted contact maps show the disappearance of the loop signal in the absence of cohesin (**Fig 2c**). This analysis confirms that the loops identified by Chromosight in WT contact maps are indeed biologically relevant structures, and not unwanted signal.

To measure more precisely the spatial distributions of cohesin loops, we computed the loop spectrum on pairs of cohesin peaks (**Fig 2d**) using Hi-C data of Hela synchronized cells released from mitosis into G1 [18]. At the beginning of the kinetics, the spectrum is flat with no significant loop scores. As cells progress through mitosis and re-enter G1, a loop signal clearly emerges with a peak

at 130 kb, a distance similar in other cell types (data not shown). The loop spectra show a secondary, weaker peak at 260 kb, suggesting a structural model with regular loop structures.

The 11,389 borders detected by Chromosight appeared enriched in CTCF deposition sites ($\simeq$40%, $P < 10^{-16}$, Fisher test) [19], showing that the detection of domain borders based on pattern matching is also relevant to capture biological features already identified with algorithms based on other approaches (ex: segmentation [20]).

Finally, Chromosight detected 1,700 hairpin like patterns (**Fig 2b**) in GM12878. Interestingly, the chromosome coordinates for this detected group are enriched in NIPBL (2 fold effect, $p < 10^{-16}$, Fisher test), a cohesin loading factor. The hairpin-like structures detected by Chromosight could therefore be interpreted as injection points for cohesin, an hypothesis of potential interest regarding the regulation of genome organization.

We analyzed Hi-C data from cells depleted of NIPBL to test for its implication in the detected hairpin patterns [21]. The hairpin patterns almost disappear in this mutant (average hairpin scores decreased from 0.40 to 0.03, $P < 10^{-16}$) (**Fig 2e**).

Discussion

Chromosight is a fast program that detects any type of pattern in chromosome contact maps for any genome. We have shown that it outperforms all other programs at reliably detecting a large number of DNA loops. In addition, it allows the user to search for any type of pattern, and additional structures could easily be added such as stripes, or patterns corresponding to large-scale structural variants (*e.g.* inversions, translocations). Chromosight could also be used to facilitate the detection of genomic misassemblies from the Hi-C signal, to help their correction and polishing [22].

Chromosight's execution time (a few minutes for human datasets) as well as its compatibility with the widely adopted `cool` format allows the exploratory analysis of large amounts of contact data. We also tested the detection of the loop, border and hairpin patterns on various animals from the DNA Zoo project [23]. The algorithm detected patterns in all tested organisms. A loop size of about 100 kb appears to be conserved across animal genomes (Supplementary Fig. 1). We envision that, as more species are investigated through Hi-C, and data resolution increases, new spatial structures will be unveiled. The user-friendly, flexible approach of Chromosight makes it a versatile tool that can easily be adapted and applied to different types of experimental data and provides a computational and statistical framework for the discovery of new principles governing chromosome architecture.

Methods

Simulation of Hi-C matrices

Simulated matrices were generated using a bootstrap strategy based on Hi-C data from mitotic *S. cerevisiae* [9] at 2kb resolution. Three main features were extracted from the yeast contact data: the probability of contact as a function of the genomic distance ($P(s)$), the positions of borders detected by HicSeg [20] and positions of loops detected manually on chromosome 5. Positions from loops and borders were then aggregated into pileups of 17x17 pixels. We generated 2000 simulated matrices of 289x289 pixels. A first probability map of the same dimension is generated by making a diagonal gradient from $P(s)$ representing the polymer matrix. For each of the 2000 generated matrices, two additional probability maps are generated. The first by placing several occurrences of the border pileup on the diagonal, where the distance between borders follows a normal distribution fitted on the experimental coordinates. The second probability map is generated by adding the loop kernel 2-100 pixels away from the diagonal with the constraint that it must be aligned vertically and horizontally with border coordinates. For each generated matrix, the product of the P(s), borders and loops probability maps is then computed and used as a probability law to sample contact positions while keeping the same number of reads as the experimental map.

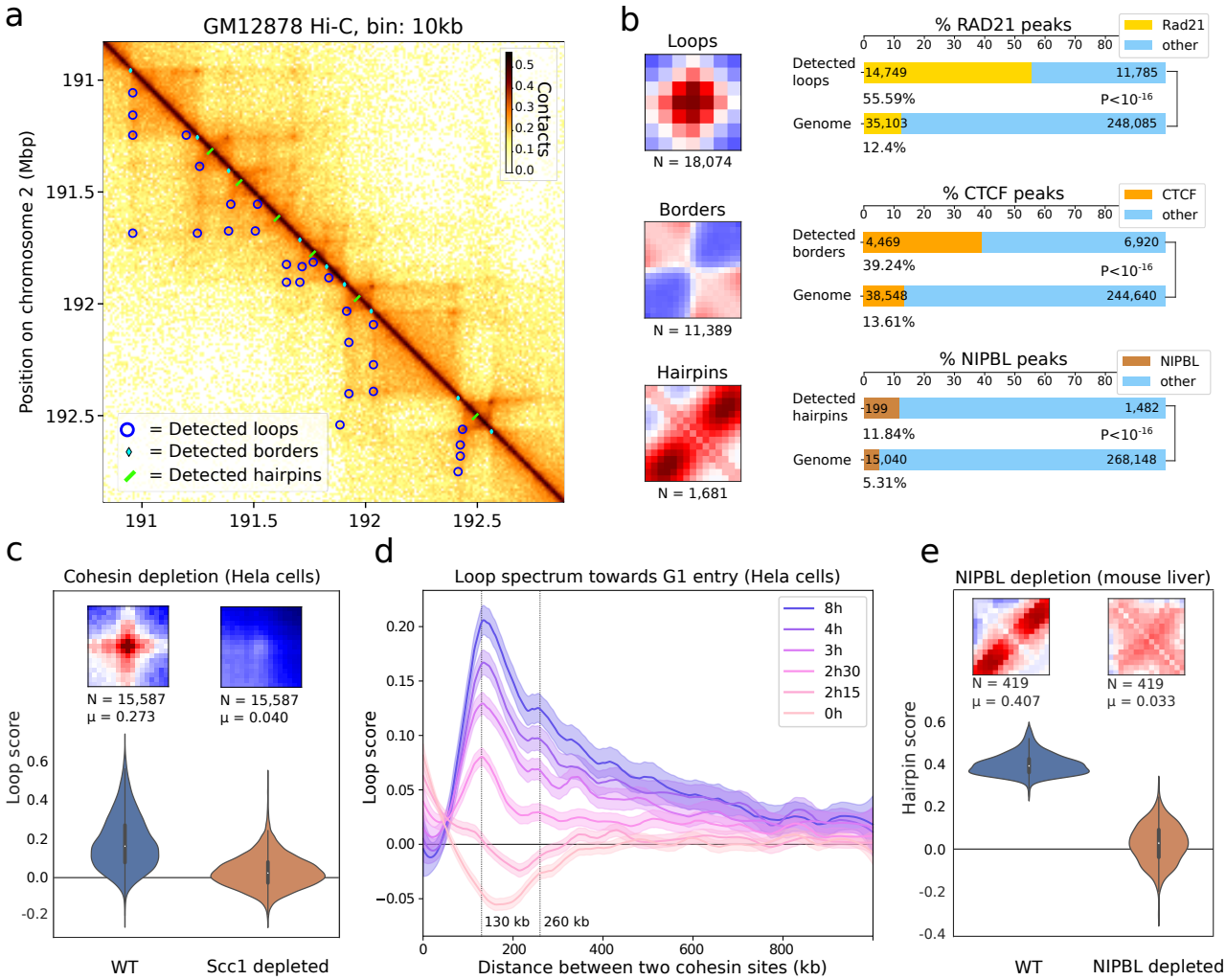

Fig. 2. Chromosight on mammalian Hi-C contact maps. a, Magnification of human chromosome 2 contact map (bin: 10 kb, [16]). The positions of the loops, borders and hairpins detected by Chromosight are indicated. **b,** Pileup (*i.e.* element-wise median) plots of windows centered on the detected loops, borders and hairpins. N: number of occurrences. Bar plots on the right panel show the enrichment in Rad21, CTCF, and NIPBL at the coordinates of loops' bases, borders, and hairpin's bases, respectively. **c,** Comparison of loop score distributions in WT (*Homo sapiens*) and in mutant cells depleted in Scc1 [17] for loops detected in WT condition. Associated pileup plots of windows centered on detected loops in WT condition. μ: mean of Pearson coefficients. **d,** Loop spectra of the loop scores for pairs of Rad21 deposition sites separated by increasing distances, at different time points during release from mitosis into G1 [18]. **e,** Comparison of hairpin score distributions in WT (*Mus musculus*, liver cells) and in mutant cells depleted in NIPBL [21] for hairpins detected in the WT condition. Associated pileup plots of windows centered on detected hairpins in WT condition.

Benchmarking

To benchmark precision, recall and F1 score, the simulated Hi-C dataset with known loop coordinates were used. Each algorithm was run with a range of 60-180 parameter combinations (Supplementary Fig. 1) on the 2000 simulated matrices and F1 score was calculated on the ensemble of results for each parameter combination separately (Supplementary Table 1). For each software, scores used in the final benchmark (Fig. 1) are those from the parameter combination that yielded the highest F1 score.

For the performance benchmark, HiCCUPS and HOMER were excluded. The former because it runs on GPU, and the latter because it uses genomic alignments as input and is much slower. The dataset used is a published high coverage Hi-C library [16] from human lymphoblastoid cell lines (GM12878). To compare RAM usage across programs, this dataset was downsampled at 10, 20, 30, 40 and 50% contacts and the maximum scanning distance was set to 2Mbp. To compare CPU time, all programs were run on the full dataset, at different maximum scanning distances, with a minimum scanning distance of 0 and all other parameters left to default. All programs were run on a single thread, on a Intel(R) Core(TM) i7-8700K CPU at 3.70GHz with 32GB of available RAM.

Software versions used in the benchmark are chromosight v0.9.0, hicexplorer v3.3.1, cooltools v0.2.0, homer 4.10 and hiccups 1.6.2.

Preprocessing of Hi-C matrices

Prior to detection, Chromosight balances the whole genome matrix using the ICE algorithm [24] to account for Hi-C associated biases. For each intrachromosomal matrix, the observed/expected contact ratios are then computed by dividing each pixel by the mean of its diagonal. This erases the diagonal gradient due to the power-law relationship between genomic distance and contact probability, thus emphasizing local variations.

Calculation of Pearson coefficients

The contact map can be considered an image $\textsc{img}_{\text{CONT}}$ where the intensity of each pixel $\textsc{img}_{\text{CONT}}[i,j]$ represents the contact probability between loci i and j of the chromosome. In that context, each pattern of interest can be considered a template image $\textsc{img}_{\text{TMP}}$ with M_{TMP} rows and N_{TMP} columns.

The correlation operation consists in sliding the template ($\textsc{img}_{\text{TMP}}$) over the image ($\textsc{img}_{\text{CONT}}$) and measuring, for each template position, the similarity between the template and its overlap in the image. We used the Pearson correlation coefficient as a the measure of similarity between the two images. The output of this matching procedure is an image of correlation coefficients $\textsc{img}_{\text{CORR}}$ such that

$$\textsc{img}_{\text{CORR}}[i,j] = Corr\left(\textsc{img}_{\text{CONT}}[i : i + M_{\text{TMP}}, j : j + N_{\text{TMP}}],\ \textsc{img}_{\text{TMP}}\right) \tag{1}$$

where the correlation operator $Corr(\cdot,\cdot)$ is defined as

$$Corr\left(\textsc{img}_{\text{X}}, \textsc{img}_{\text{Y}}\right) = \frac{cov(\textsc{img}_{\text{X}}, \textsc{img}_{\text{Y}})}{std(\textsc{img}_{\text{X}}) \cdot std(\textsc{img}_{\text{Y}})} \tag{2}$$

$$= \frac{\sum_i \sum_j (\textsc{img}_{\text{X}}(i,j) - \overline{\textsc{img}_{\text{X}}}) \cdot (\textsc{img}_{\text{Y}}(i,j) - \overline{\textsc{img}_{\text{Y}}})}{\sqrt{\sum_i \sum_j (\textsc{img}_{\text{X}}(i,j) - \overline{\textsc{img}_{\text{X}}})^2} \cdot \sqrt{\sum_i \sum_j (\textsc{img}_{\text{Y}}(i,j) - \overline{\textsc{img}_{\text{Y}}})^2}} \tag{3}$$

where $\overline{\textsc{img}} = \frac{1}{M \cdot N} \sum_i \sum_j \textsc{img}(i,j)$.

Separation of high-correlation foci

Selection is done by localizing specific local maxima within $\textsc{img}_{\text{CORR}}$. We proceeded as follows: first, we discard all points (i,j) where $\textsc{img}_{\text{CORR}}[i,j] < \tau_{\text{CORR}}$. An adjacency graph A_{dxd} is then generated from the d remaining points. The value of $A[i,j]$ is a boolean indicating the (4-way) adjacency status

between the ith and j^{th} nonzero pixels. The scipy implementation of the CCL algorithm for sparse graphs is then used on D to label the different contiguous foci of nonzero pixels. Foci with less than two pixels are discarded. For each focus, the pixel with the highest coefficient is determined as the pattern coordinate.

Patterns are then filtered out if they overlap too many empty pixels or are too close from another detected pattern. The remaining candidates in IMG_{CORR} are scanned by decreasing order of magnitude: every time a candidate is appended to the list of selected local maxima, all its neighboring candidates are discarded. The proportion of empty pixels allowed and the minimum separation between two patterns are also user defined parameters.

Author contributions

All authors contributed to the design of the algorithm. CMD, AB, LB, AC implemented it. CMD, NG, RM compared Chromosight to other programs. LB and AC designed the data simulations. CMD, PM, RK and AC analysed biological data and interpret results. CMD, RK and AC wrote the manuscript.

Competing financial interests

The authors declare no competing financial interests.

Acknowledgements

This work was initiated as a hackathon challenge regrouping Institut Pasteur scientists and ENGIE engineers. We thank all the people that made this event possible, especially Anne-Gaëlle Coutris, Romain Tchertchian and Olivier Gascuel. Frédéric Beckouët and all the members of Spatial Regulation of Genomes unit are thanked for stimulating discussions and feedbacks. AB works within the framework of a "Mécénat Compétence" contract of the company ENGIE. CMD is supported by the Pasteur - Paris University (PPU) International PhD Program. NG is supported by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement 764840. This research is supported by funding from Agence Nationale de la Recherche (ANR JCJC 2019, "Apollo") to AC and funding to RK from the European Research Council under the Horizon 2020 Program (ERC grant agreement 260822).

References

[1] Job Dekker, Karsten Rippe, Martijn Dekker, and Nancy Kleckner. Capturing chromosome conformation. *Science*, 295:1306–1311, 2002.

[2] Erez Lieberman-Aiden, Nynke L. van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R. Lajoie, Peter J. Sabo, Michael O. Dorschner, Richard Sandstrom, Bradley Bernstein, M. A. Bender, Mark Groudine, Andreas Gnirke, John Stamatoyannopoulos, Leonid A. Mirny, Eric S. Lander, and Job Dekker. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, 326(5950):289–293, Oct 2009.

[3] Suhas S P Rao, Su-Chen Huang, Brian Glenn St Hilaire, Jesse M Engreitz, Elizabeth M Perez, Kyong-Rim Kieffer-Kwon, Adrian L Sanborn, Sarah E Johnstone, Gavin D Bascom, Ivan D Bochkov, Xingfan Huang, Muhammad S Shamim, Jaeweon Shin, Douglass Turner, Ziyi Ye, Arina D Omer, James T Robinson, Tamar Schlick, Bradley E Bernstein, Rafael Casellas, Eric S Lander, and Erez Lieberman Aiden. Cohesin loss eliminates all loop domains. *Cell*, 171(2):305–320.e24, October 2017.

[4] Elphège P Nora, Bryan R Lajoie, Edda G Schulz, Luca Giorgetti, Ikuhiro Okamoto, Nicolas Servant, Tristan Piolot, Nynke L van Berkum, Johannes Meisig, John Sedat, Joost Gribnau, Emmanuel Barillot, Nils Blüthgen, Job Dekker, and Edith Heard. Spatial partitioning of the regulatory landscape of the x-inactivation centre. *Nature*, 485(7398):381–5, April 2012.

[5] Wouter de Laat and Denis Duboule. Topology of mammalian developmental enhancers and their regulatory landscapes. *Nature*, 502(7472):499–506, October 2013.

[6] Jesse R. Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S. Liu, and Bing Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398):376–380, May 2012.

[7] Elzo de Wit, Erica S M Vos, Sjoerd J B Holwerda, Christian Valdes-Quezada, Marjon J A M Verstegen, Hans Teunissen, Erik Splinter, Patrick J Wijchers, Peter H L Krijger, and Wouter de Laat. Ctcf binding polarity determines chromatin looping. *Mol. Cell*, 60(4):676–84, November 2015.

[8] Daniel M. Ibrahim and Stefan Mundlos. Three-dimensional chromatin in disease: What holds us together and what drives us apart? *Current Opinion in Cell Biology*, 64:1 – 9, 2020.

[9] Lise Dauban, Rémi Montagne, Agnès Thierry, Luciana Lazar-Stefanita, Nathalie Bastié, Olivier Gadal, Axel Cournac, Romain Koszul, and Frédéric Beckouët. Regulation of cohesin-mediated chromosome folding by eco1 and other partners. *Mol. Cell*, January 2020.

[10] Suhas S P Rao, Miriam H Huntley, Neva C Durand, Elena K Stamenova, Ivan D Bochkov, James T Robinson, Adrian L Sanborn, Ido Machol, Arina D Omer, Eric S Lander, and Erez Lieberman Aiden. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–80, December 2014.

[11] Aaron T L Lun and Gordon K Smyth. diffhic: a bioconductor package to detect differential genomic interactions in hi-c data. *BMC Bioinformatics*, 16:258, August 2015.

[12] Joachim Wolff, Rolf Backofen, and Bjoern Gruening. Loop detection using hi-c data with hicexplorer. *bioRxiv*, 2020.

[13] Sven Heinz, Lorane Texari, Michael G B Hayes, Matthew Urbanowski, Max W Chang, Ninvita Givarkes, Alexander Rialdi, Kris M White, Randy A Albrecht, Lars Pache, Ivan Marazzi, Adolfo García-Sastre, Megan L Shaw, and Christopher Benner. Transcription elongation can affect genome 3d structure. *Cell*, 174(6):1522–1536.e22, 09 2018.

[14] Nezar Abdennur and Leonid A Mirny. Cooler: scalable storage for Hi-C data and other genomically labeled arrays. *Bioinformatics*, 07 2019.

[15] Roberto Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley Publishing, 2009.

[16] Jay Ghurye, Arang Rhie, Brian P Walenz, Anthony Schmitt, Siddarth Selvaraj, Mihai Pop, Adam M Phillippy, and Sergey Koren. Integrating hi-c links with assembly graphs for chromosome-scale assembly. *PLoS Comput. Biol.*, 15(8):e1007273, 08 2019.

[17] Gordana Wutz, Csilla Várnai, Kota Nagasaka, David A Cisneros, Roman R Stocsits, Wen Tang, Stefan Schoenfelder, Gregor Jessberger, Matthias Muhar, M Julius Hossain, Nike Walther, Birgit Koch, Moritz Kueblbeck, Jan Ellenberg, Johannes Zuber, Peter Fraser, and Jan-Michael Peters. Topologically associating domains and chromatin loops depend on cohesin and are regulated by ctcf, wapl, and pds5 proteins. *EMBO J.*, 36(24):3573–3599, 12 2017.

[18] Kristin Abramo, Anne-Laure Valton, Sergey V Venev, Hakan Ozadam, A Nicole Fox, and Job Dekker. A chromosome folding intermediate at the condensin-to-cohesin transition during telophase. *Nat. Cell Biol.*, 21(11):1393–1402, 11 2019.

[19] The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012.

[20] Celine Lévy-Leduc, Maud Delattre, Tristan Mary-Huard, and Stephane Robin. Two-dimensional segmentation for analyzing hi-c data. *Bioinformatics*, 30(17):i386–i392, 2014.

[21] Wibke Schwarzer, Nezar Abdennur, Anton Goloborodko, Aleksandra Pekowska, Geoffrey Fudenberg, Yann Loe-Mie, Nuno A Fonseca, Wolfgang Huber, Christian H Haering, Leonid Mirny, and Francois Spitz. Two independent modes of chromatin organization revealed by cohesin removal. *Nature*, 551(7678):51–56, 11 2017.

[22] Hervé Marie-Nelly, Martial Marbouty, Axel Cournac, Jean-François Flot, Gianni Liti, Dante Poggi Parodi, Sylvie Syan, Nancy Guillén, Antoine Margeot, Christophe Zimmer, and Romain Koszul. High-quality genome (re)assembly using chromosomal contact data. *Nat Commun*, 5:5695, 2014.

[23] Olga Dudchenko, Sanjit S. Batra, Arina D. Omer, Sarah K. Nyquist, Marie Hoeger, Neva C. Durand, Muhammad S. Shamim, Ido Machol, Eric S. Lander, Aviva Presser Aiden, and Erez Lieberman Aiden. De novo assembly of the aedes aegypti genome using hi-c yields chromosome-length scaffolds. *Science*, 356(6333):92–95, 2017.

[24] Maxim Imakaev, Geoffrey Fudenberg, Rachel Patton McCord, Natalia Naumova, Anton Goloborodko, Bryan R Lajoie, Job Dekker, and Leonid A Mirny. Iterative correction of hi-c data reveals hallmarks of chromosome organization. *Nature methods*, 9(10):999–1003, 2012.

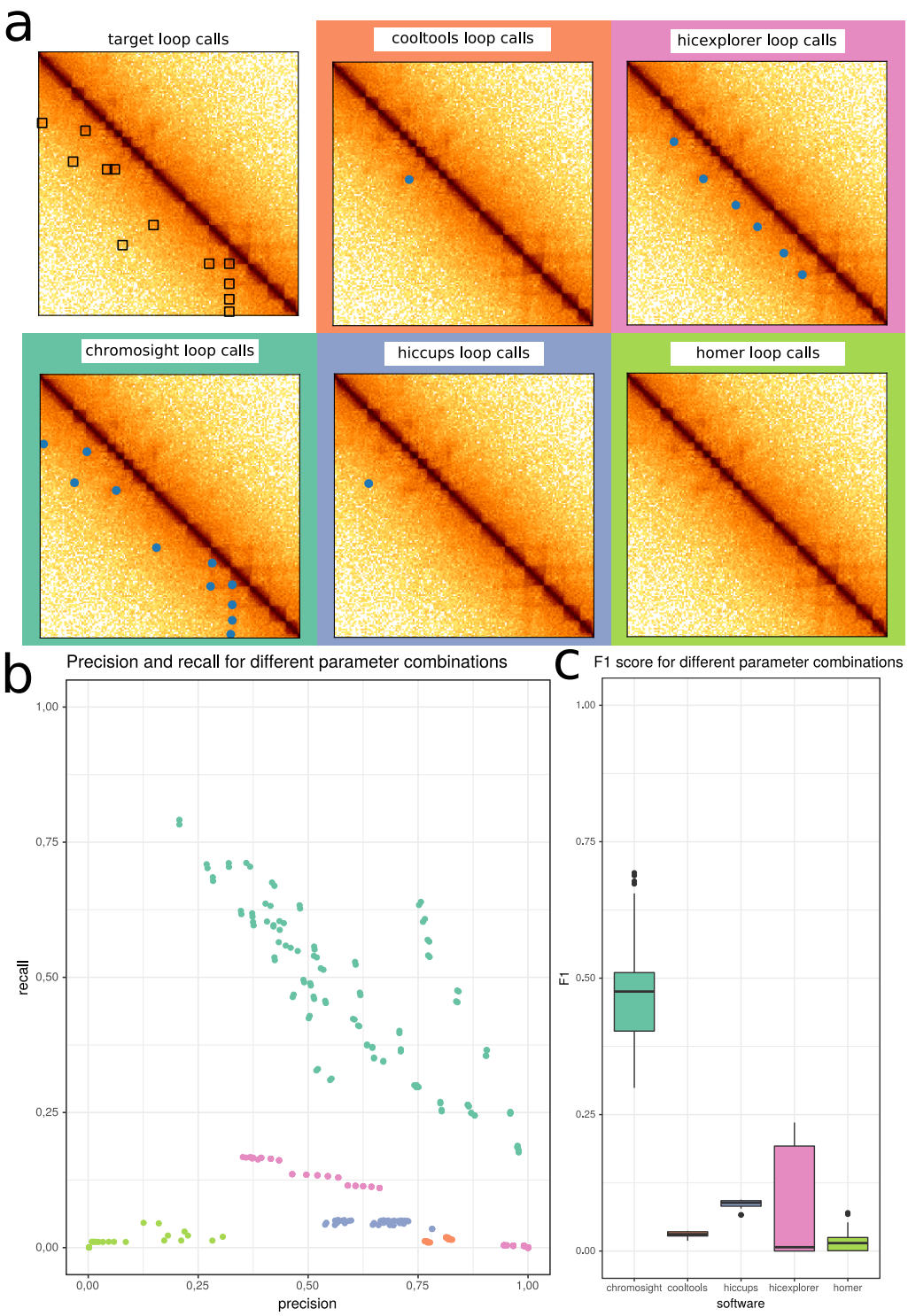

Fig S1 **Benchmark results.** Detailed benchmark results. a) Example region from a synthetic matrix with real loop calls (top left) and loops detected by all algorithms used in the benchmark. b) Precision and recall from all algorithms on synthetic matrices, on the whole range of parameters tested. c) Distribution of F1 Scores for each algorithm for the whole range of parameters.

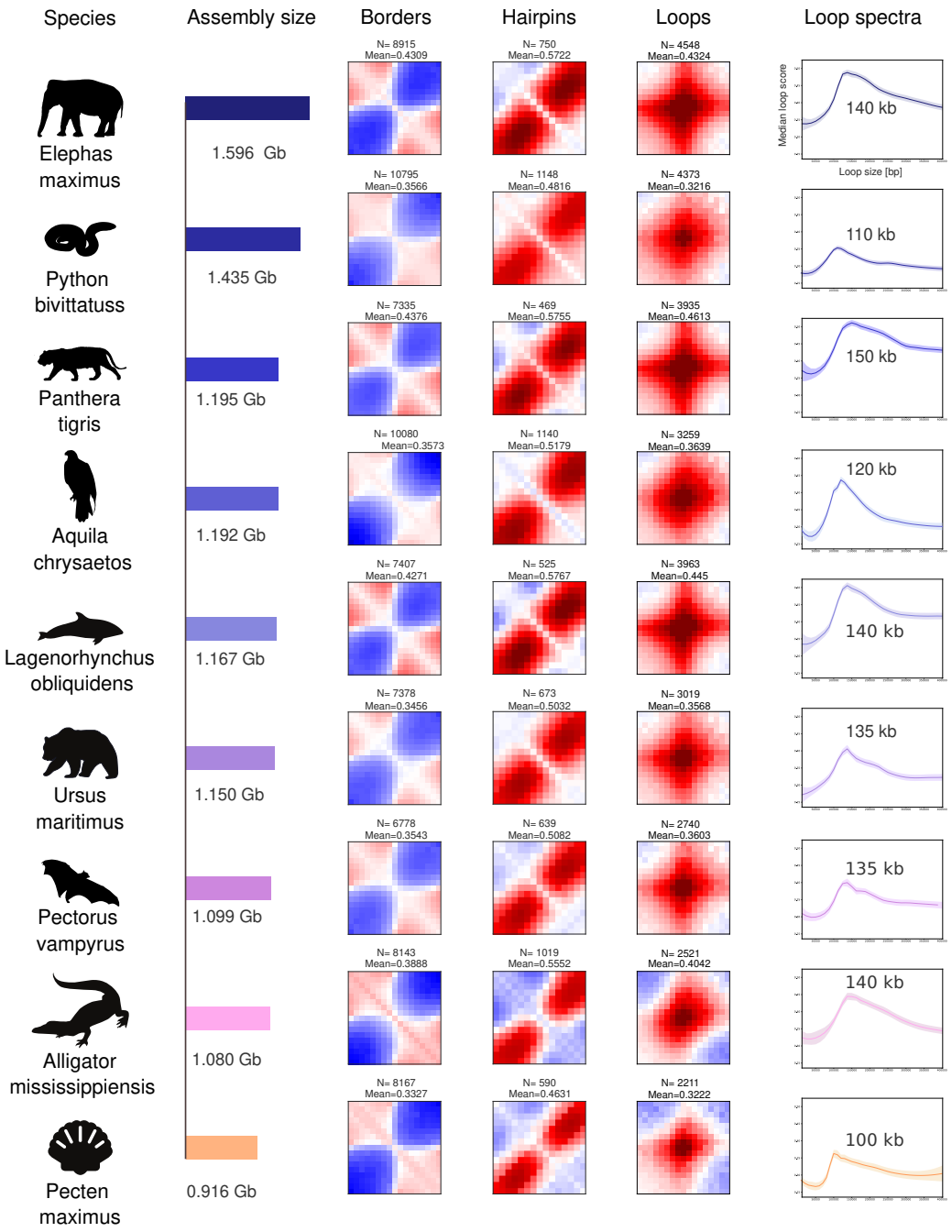

Fig S2 **Detection of loops, borders, hairpins in various animals from the DNA zoo project** [23] From left to right, name of the species, barplot representing the length of the DNA zoo genome assembly, associated pileup plots for detected border, hairpin and loop patterns and loop spectra computed on the positions of detected loops. The loop spectrum gives the size at which the detected loops have the highest scores. The detections have been performed on 12.5kb matrices.

software	parameter	values
chromosight	–window-size	10,15,20
chromosight	–min-dist	0,40000
chromosight	–pearson	0.30,0.35,0.40,0.45,0.50
chromosight	–min-separation	0,50000
chromosight	–full	On,off
hicexplorer	–windowSize	10,15,20
hicexplorer	–peakWidth	4,5,6,7,8
hicexplorer	–peakInteractionsThreshold	10,20,30
hicexplorer	–pValuePreselection	0.01,0.02,0.05,0.1
cooltools	–max-loci-separation	100000,200000,1000000,2000000
cooltools	–max-nans-tolerated	5,10,15,20
cooltools	–dots-clustering-radius	14000,19000,34000,39000
hiccups	-p	1,2,4,6
hiccups	-i	6,10,14
hiccups	-f	0.05,0.1,0.2
homer	-poissonLoopGlobalBg	0.0001,0.001
homer	-poissonLoopLocalBg	0.01,0.05,0.1
homer	-window	2000,5000,10000

Table S1 **Parameters used in the benchmark.** Name and value of all parameters used in the benchmark for each software.