Statistically rigorous and computationally efficient chromatin stripe detection with Quagga

Fan Feng, Sean Moran, Anders Hansen, et al.

Genome Res. published online November 12, 2025
Access the most recent version at doi:10.1101/gr.280132.124

Statistically Rigorous and Computationally Efficient Chromatin Stripe Detection with Quagga

Fan Feng[1,†], Sean Moran[1,†], Anders S. Hansen[2,3,4], Xiaotian Zhang[5*], Jie Liu[1,6*]

[1]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, 48103, USA
[2]Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, 02139, USA
[3]Gene Regulation Observatory, Broad Institute of MIT and Harvard, Cambridge, MA, 02139, USA
[4]Koch Institute for Integrative Cancer Research, Cambridge, MA, 02139, USA
[5]Department of Biochemistry and Molecular Biology, University of Texas Health Science Center Houston, 6431 Fannin St, Houston, TX, USA
[6]Department of Computer Science and Engineering, University of Michigan, Ann Arbor, 48103, MI, USA

[*]Corresponding authors. E-mail: Xiaotian.Zhang@uth.tmc.edu; drjieliu@umich.edu
Contributing authors: fanfeng@umich.edu; spmoran@umich.edu; ashansen@mit.edu;
[†]These authors contributed equally to this work.

Abstract

Chromatin stripes are architectural chromatin features where a singular loop anchor interacts with a contiguous region of DNA so, at the bulk sequencing level, it appears as a long stripe on chromatin contact matrices. Stripes are thought to play an important role in gene regulation and have been implicated in regulating a cell's lineage determination. Therefore, integrated analysis of stripes with genomic and epigenomic features at a genome-wide scale shows vast potential in understanding the cooperation between regulatory elements in 3D nucleome. To this end, we present Quagga, a computational tool for detection and statistical verification of genomic architectural stripes from Hi-C or Micro-C chromatin contact maps, which relies on robust image processing techniques and rigorous statistical tests for enrichment. Quagga outperforms other stripe detection methods in accuracy and is highly versatile, working with Hi-C, Micro-C, and other chromatin conformation capture data. By reporting on all tools' performance in classifying CTCF-cohesin anchored stripes, enhancer-promoter interacting stripes, and indeterminate stripes, we also demonstrate a thorough, integrated analysis to determine the output stripes' quality. Our work provides a flexible and convenient tool to help scientists explore the relationships between chromatin architectural stripes and important biological questions.

Introduction

Chromatin conformation capture (3C) techniques, especially proximity ligation-based methods, have revealed that the hierarchical structures of DNA break down into distinct architectural features; these include A/B compartments, topologically associating domains (TADs), and chromatin loops (Dixon et al., 2012; Lieberman-Aiden et al., 2009; Nora et al., 2012; Rao et al., 2014; Sexton et al., 2012). Improvements in 3C sequencing technologies resulted in improved sequencing depth, and higher-resolution contact maps generated by *in situ* Hi-C have made it possible to study architectural stripes (Vian et al., 2018). Chromatin stripes are a chromatin architectural feature that captures the dynamic, ever-changing nature of the genome. The stripes reflect interactions between a single locus (stripe anchor) and a continuum of genomic regions (Banigan et al., 2020). For example, chromatin stripes may be based on CTCF/cohesin loops, where multiple genomic sites that lie far apart linearly

are brought into spatial proximity to a distal single locus by loop extrusion. At the bulk-sequencing scale, a corresponding chromatin stripe is a place where multiple loop configurations are anchored to one particular part of the genome, and their "sliding states" are affected at the population level, giving them a stripe appearance on the chromatin contact map, which reflects the highly dynamic nature of chromatin looping (Davidson et al., 2019; Gabriele et al., 2022). An integrated analysis of stripes with genomic and epigenomic features at a genome-wide scale also shows vast potential in understanding the cooperation between regulatory elements in 3D space (Barrington et al., 2019; Kraft et al., 2019; Zhang et al., 2019; Hsieh et al., 2020).

Previous algorithms take diverse strategies in calling stripes. Chromosight (Matthey-Doret et al., 2020) defines kernel matrices, and applies a convolution-like approach to scan the contact map to identify the loops. However, the pre-defined kernel is not extensively benchmarked, not robust to various resolutions and stripe sizes, and the method also lacks a reasonable thresholding function to distinguish stripes from non-stripe noises. Zebra (Vian et al., 2018) uses Poisson statistics to identify pixels with significantly enriched contacts, which is commonly applied to Hi-C in loop callers (Rao et al., 2014). Nevertheless, this approach is likely to call TAD boundaries or loops as false-positive stripes, and previous research studying stripes via Zebra (Vian et al., 2018) needed to manually remove TADs from their stripe list. Stripenn (Yoon et al., 2022) first uses image processing techniques, including a Gaussian filter and Canny edge detection, to identify candidate stripes. Then p-values and stripiness are calculated for each candidate stripe to distinguish the significant ones. However, stripiness is an arbitrary metric that is weaker in statistical power. Stripenn also does not accurately assess the length of stripes, always calculating stripe starting position as directly on the main diagonal. StripeDiff (Gupta et al., 2022) detects differential stripes between experiments and reveals the connection between changes of chromatin stripe and chromatin modification, transcriptional regulation, and cell differentiation. Nevertheless, StripeDiff does not provide a user-friendly package to directly call all stripes with common contact map files.

Here we introduce Quagga, a statistically rigorous, algorithmically efficient, and interpretable tool to identify stripes. Quagga's biggest innovation lies in its statistical rigor coupled to computational efficiency. Quagga samples pixel levels across all candidate stripe regions, resulting in the calculation

of millions of *p*-values, which is original in its dual use of combining calculating stripe significance with solving for stripe length. Quagga's thorough testing ensures the reliability of stripe calls, particularly in complex regions where other structural patterns, including TAD boundaries, may be called as false positives. Notably, Quagga caches its statistical testing calculations, allowing it to sample a very large volume of candidate stripes in a competitive amount of time. Finally, Quagga's 3C data filtering and peak finding are done in a straightforward way with Gaussian and Gabor kernel filtering; this approach is interpretable and simple, allowing any user to easily fine-tune Quagga to their needs.

Quagga has been extensively tested on Hi-C and Micro-C, and is adaptable to use on any 3C-family matrix in HIC or COOL format. We report our comparison of Quagga against two gold-standard stripe callers, Zebra and Stripenn, we show off CTCF-type stripe detection and its minimizing of off-target stripes. We also analyzed each tool's ability to recall enhancer-promoter (EP) interacting stripes. Quagga is freely available on GitHub and makes possible the detection and rigorous analysis of stripes in a lightweight and adaptable manner.

Results

Quagga overview

Quagga is a chromatin stripe detection tool that determines chromosomal coordinates that form an accurate bounding box around detected chromatin stripes. It works on all 3C family sequencing types, including Hi-C, Micro-C, and HiChIP. Quagga uses two major steps to find chromatin stripes. First, Quagga performs a quick screening of potential stripes and determines stripe width. Second, using the potential stripe list, Quagga checks for statistical enrichment and determines the stripe's length (Fig. 1A).

Quagga uses signal processing techniques to perform naive stripe calling and determine width. First, it reduces the chromosome into 5' or 3' signal vectors. A vertical or horizontal loading matrix is filled, taking a maximum length along the 5' or 3' direction on the contact frequency matrix. For example, a horizontal loading matrix has its rows containing the values of the rows of the corresponding coordinates of the contact frequency matrix, starting from the main diagonal and out to the maximum distance specified. A vertical loading matrix is filled similarly, except its rows with the values of the

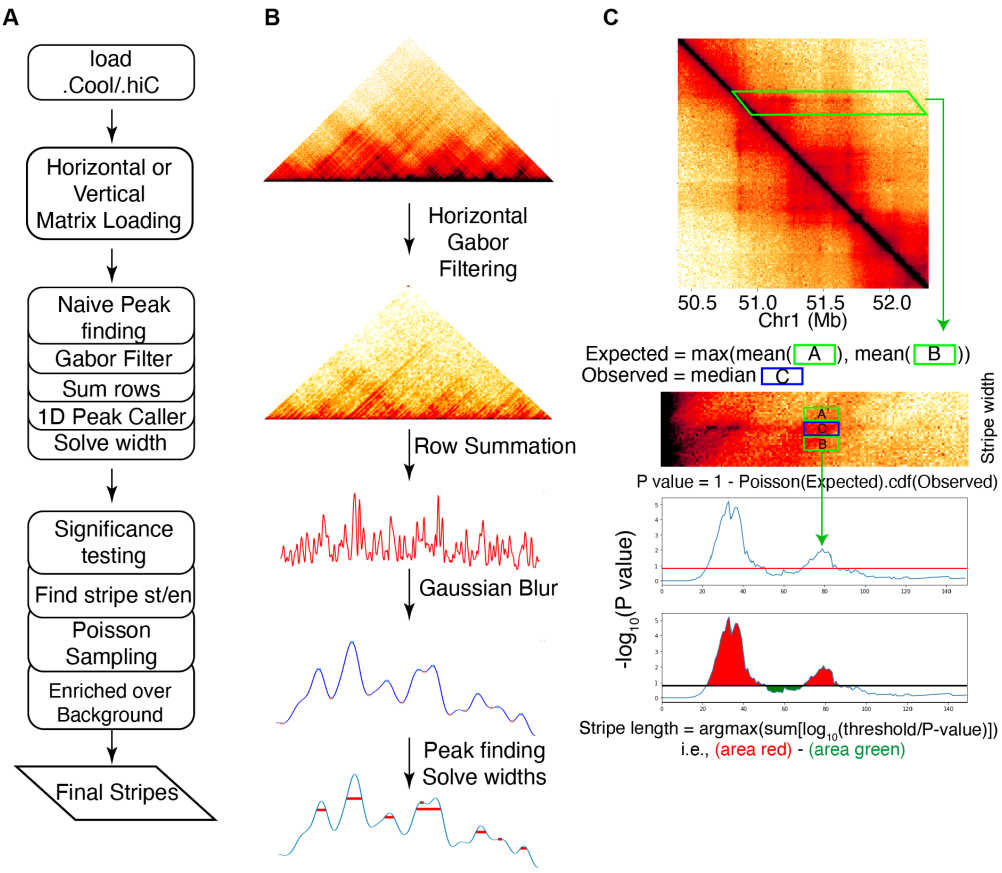

Figure 1: Quagga is a lightweight tool that calls architectural stripes from 3C chromatin contact frequency matrix. (A) The workflow of Quagga. (B) 3C family-type contact frequency matrix via a COOL or HIC file is processed into a horizontally or vertically loaded matrix, and a naive peak finding algorithm is used; stripe indices and width are calculated based on the vertically or horizontally averaged row or column sums. (C) Significance testing is applied to the candidate stripes from the previous step. Called region windows are used to determine the appropriate length of the stripe and whether or not the stripe is enriched over the local background.

columns from the contact frequency matrix instead. Then Quagga filters the loader matrices using a Gabor kernel Granlund (1978) to focus the angular signal from the underlying image. This strategy emphasizes high-contrast lines instead of dots or spots, which reduces noise and avoids single loops being called. Then the loader matrices are summed over their rows to form a 1-D signal vector that reflects the signal buildup over its respective axis; the intuition is that major anchors will accumulate a tremendous amount of contact frequency signal and that non-stripe loop or TAD interactions can be filtered out in statistical enrichment. We apply Gaussian blur to reduce the noise of the vector and run a 1-D peak-finding algorithm to determine local maxima and their widths (Fig. 1B) as our naive stripes.

Quagga next checks all stripes to determine appropriate stripe calls and assigns stripes a length. To do this, Quagga calculates a *p*-value for each pixel along the candidate stripes based on Poisson statistics by comparing observed to expected chromatin contact frequency (Fig. 1C). Observed is determined along the length of the naive stripe using the solved width; expected values are based on windows flanking the stripe area. Using the *p*-values solved along the length of the stripe, Quagga uses a simple maximization calculation to determine the most likely starting and ending positions. In this way, an integrity score like Stripenn's *stripiness* is not necessary as Quagga already checks for stripe integrity by maximizing the significance of the observed stripe over the expected signal based on its neighbors.

Quagga efficiently detects stripes across diverse cell types, sequencing depths, resolutions, and 3C methods

We applied Quagga to call stripes from Hi-C contact maps of different cell lines. Quagga found 4,133 stripes for GM12878 at 10 kb resolution, 10,398 for H1, 2,908 for K562, and 8,189 for HFFc6, respectively (Supplementary Fig. 1). Similar to Hi-C loop callers, Quagga calls are also dependent on sequencing depth and resolution. On a standard Hi-C GM12878 dataset (Rao et al., 2014), we tested a range of sequencing depths from 4 billion filtered read pairs to as low as 15.6 million filtered read pairs, and found Hi-C data stripe detection effective at 250 million filtered reads or more (Supplementary Fig. 2-3). Quagga was consistent between 5 kb and 10 kb resolution calls; Quagga identified fewer stripes at 5 kb resolution, and these stripes also tended to be shorter in length. We believe this is due to increased sparsity in the contact maps at higher resolution (Supplementary Fig. 4). In particular, longer-range interactions become even sparser due to distance-dependent contact decay, making it less likely for stripes to be detected over longer genomic distances.

Quagga's stripe calls were also consistent between biological replicates. Of stripes found by Quagga, 673 of 1167 were unique in bioreplicate 1 (Rep1) and 344 of 837 were unique in bioreplicate 2 (Rep2), which for shared stripes between the two replicates, is 42.3 and 58.9 percent, respectively (Supplementary Fig. 5A).

Quagga can also be applied to additional chromatin conformation capture data, such as HiChIP.

We found 3,782 stripes from HiChIP in SMC1 and 2,970 stripes in CTCF (Supplementary Table 1). We also attempted to call Quagga on split-pool recognition of interactions by tag extension data (SPRITE); however, due to the differences in the underlying chemistry, Quagga required hand annotation to find true positive stripes (Supplementary Fig. 6).

Quagga is also algorithmically efficient. For stripe calling from GM12878 Hi-C data, Quagga required 99 minutes, compared to 65 minutes for Zebra and 505 minutes for Stripenn. In all tools, the primary time-consuming step is statistical significance testing, as more stripe candidates lead to more significance assessments. Despite performing substantially more statistical tests, Quagga completes its runs within a reasonable timeframe.

Stripes called by Quagga have the most variety and match size-scale

We demonstrate the ability of Quagga to detect stripes from Hi-C contact maps by aggregating the various sizes of all called stripes of Quagga, Stripenn, and Zebra. These stripes are based on the same publicly available GM12878 Hi-C data at 10 kb resolution calls on the whole genome. Quagga called 4,133 stripes, Zebra called 3,278 stripes, and Stripenn called 873 stripes (Fig. 2A). A considerable number of each tool's stripes are unique, but many are also shared (Supplementary Fig. 7). Looking at tool-exclusive stripes, Quagga found 2,702, Zebra 1,826, and Stripenn 169 stripes, where Quagga-only stripes have the highest enrichment of CTCF binding (Supplementary Fig. 8). To assess how reasonable the aggregates of each tool's stripe output were, we quantitatively examined the differences in their distribution of stripe lengths, stripe widths, and performed two kinds of pileup aggregated peak analysis (APA). We defined a reasonable distribution of peak lengths as one near the median CTCF/Cohesin loop anchor span (300 kb) and at or below the median TAD sizes (500 kb) (Dixon et al., 2012; Fudenberg et al., 2016; Xi and Beer, 2021). Quagga stripe lengths range mostly between 50 kb to 320 kb. Zebra's total span trended longer than Quagga's overall, but enough of the lengths appeared to be reasonable; while Stripenn's stripe lengths are normally distributed around 1 Mb long (Fig. 2A). These results suggest that Quagga and Zebra are more sensitive to detecting shorter stripes than Stripenn. And although Zebra has a reasonable length distribution, there are a large number of false positive stripes if we do not perform the additional steps of removing loops

and TAD boundaries, with example visualized in Fig. 2E and Supplementary Fig. 9. Stripe width distributions between Quagga, and Stripenn highlight Quagga's adaptability to capture many different widths in comparison to Stripenn (Fig. 2B). Zebra only provides the midpoint of the solved stripe coordinate, as all those stripes are uniformly reported as 10 kb wide.

Aggregated peak analysis pile-up using stripe coordinates on the source Hi-C data (Fig. 2C) demonstrates the uniformity of stripe calls in Stripenn, while showing that a variety of signal widths and lengths are in Quagga and Zebra, evident by the tapering pileup. This analysis underlines Quagga's versatility in comparison to the other tools, evident by the longer, wider, and non-uniform pileup that captures a very wide variety of stripe lengths and widths. Tool-unique stripe APAs demonstrate Stripenn has a stark, block appearance, along with Stripenn's high count of long stripes, suggesting many counts may be topologically associated domains (Supplementary Fig. 8). Finally, a dynamically scaled APA pileup along the length of stripes per tool shows the relative contact signals along the stripes (Fig. 2D). As all signal is captured in these length-scaled windows, the presence of signal before the starting line is expected, as this should include the main diagonal or other elements in line with the stripe that are not captured. Stripenn always starts at the main diagonal, so no leading signal would be counted before the start point.

Quagga identifies the close connection between Hi-C stripes and CTCF-cohesin binding

The Hi-C stripe formation is commonly explained with one-sided extrusion. Therefore, we investigated the CTCF/RAD21 binding pattern of stripe anchors to assess Quagga's outputs. In the GM12878 Hi-C contact map, Quagga identified 4,133 stripes at 10 kb resolution. Quagga-identified stripes are further categorized into two groups - 3' stripes spanning downstream and 5' stripes spanning upstream. We analyzed CTCF/RAD21 enrichment and CTCF orientation patterns at Quagga-called stripes, with the binding sites identified from ChIP-seq data and CTCF orientations annotated based on the underlying DNA sequence motifs. Our analysis revealed a significant enrichment of CTCF and RAD21 at both 3' and 5' stripe anchors identified by Quagga, signifying the dominance of extrusion-based stripes (Fig. 3A-B).

We also detect distinct patterns of CTCF orientation for the two stripe groups. Specifically, the 3'

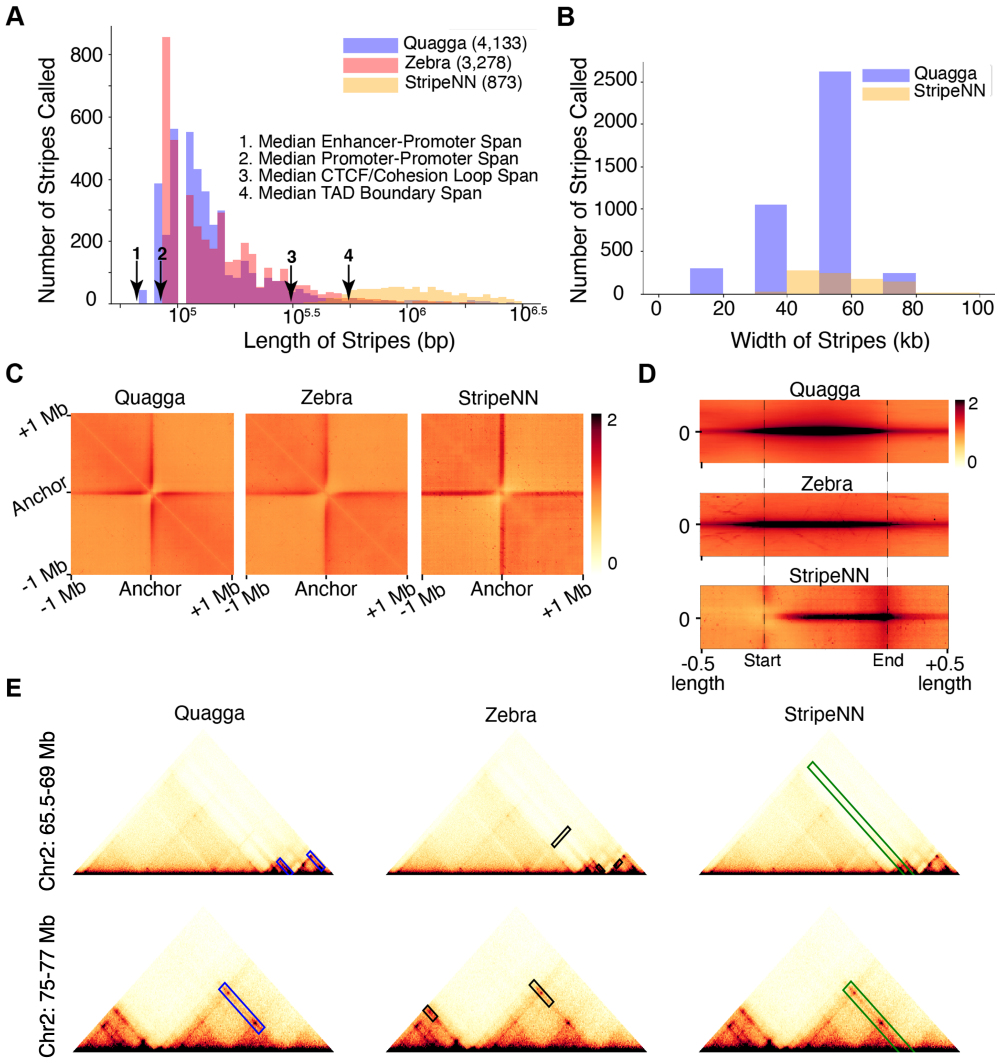

Figure 2: Benchmarking of three stripe callers, including Quagga, Zebra, and Stripenn, with Hi-C contact maps. Quagga's (A) stripe lengths lie mostly in the expected range compared to Stripenn or Zebra. (B) Stripe widths for Quagga are also the most varied. (C) Aggregated peak pileups using a 2 Mb window centered on the point of the main diagonal nearest the stripe's anchor, which demonstrates the high variability of stripe length and width Quagga captures. (D) Scaled Quagga pileups along the entire stripe length show Quagga's variability and distribution of stripes; the beginning of the stripe is fixed to the start position, and half its length in either direction is captured as a context sequence. (E) Examples of stripes called by Quagga, Zebra, and Stripenn from GM12878 Hi-C.

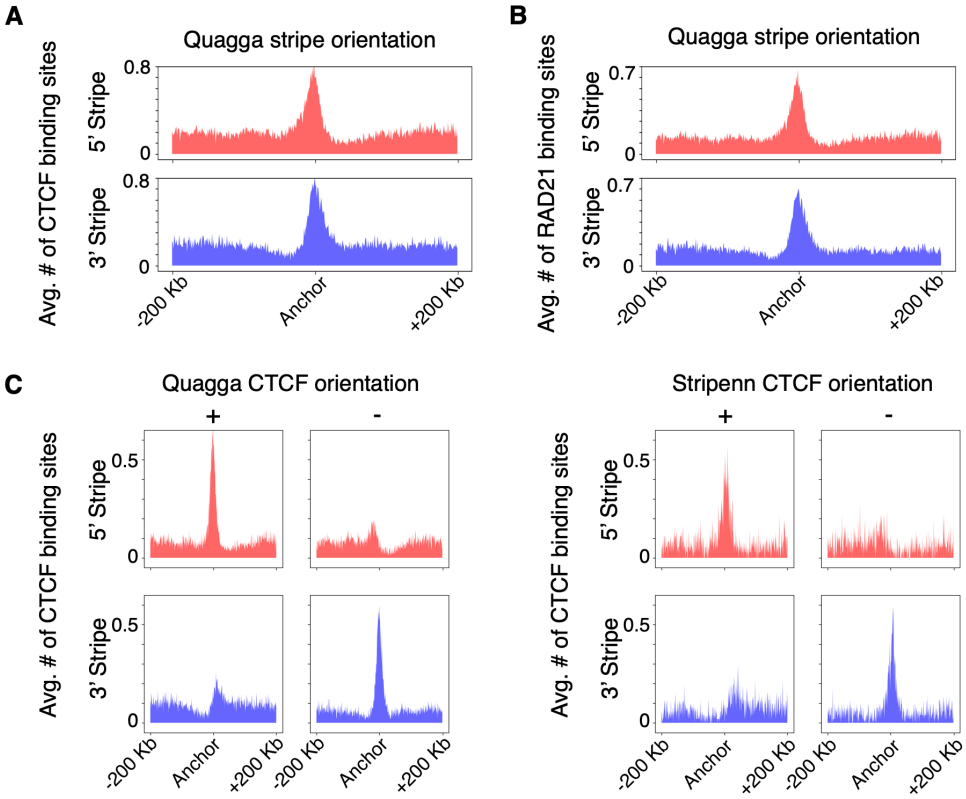

Figure 3: Hi-C stripes called by Quagga are closely related to CTCF/RAD21 extrusion. For both 3' and 5' stripes, (A) CTCF and (B) RAD21 are enriched at GM12878 stripe anchors. (C) Quagga detects four times stripes as Stripenn, and demonstrates the same CTCF enrichment pattern compared to Stripenn, in which 5' stripe anchors are more enriched in positive-strand CTCF, and the 3' stripes are more enriched in negative-strand CTCF.

stripes exhibited a significant enrichment of forward-strand (+) CTCF orientations and a depletion of backward-strand (-) CTCF orientations, with the reverse being true for the 5' stripes (Fig. 3C). As a comparison, stripes called by the baseline method Stripenn are less enriched in CTCF binding (Fig. 3C). This observation demonstrates that Quagga effectively identifies the prototypical CTCF/RAD21-extrusion stripes in Hi-C data.

Our assessment of CTCF stripe assignment is supported by performing stripe calls on Auxin-induced CTCF degradation, a technique for loop knockdown. Quagga called 3,873 stripes on the control compared to 55 in the Auxin-treated sample, strongly indicative that CTCF-loops play an outsize role in chromatin stripe formation (Supplementary Fig. 10).

Quagga calls more CTCF-stripes and has the least indeterminate stripes

To assess the quality of chromatin stripes found using Quagga, Zebra, and Stripenn, we sought to classify our stripes into three categories, namely (1) CTCF-motif occupied by CTCF stripes "CTCF-stripes", (2) CTCF-deficient stripes that intersect with EP-interacting regions "EP-stripes", and (3) CTCF-deficient stripes that are not EP-intersecting "indeterminate". We follow the convention of previous studies in simplifying promoter-promoter and enhancer-promoter interactions both as EP-interactions (Hsieh et al., 2022). From 4,133 Quagga stripe calls, 87.83% were CTCF-occupied (3,630), 2.6% were EP-stripes (110), and 9.5% indeterminate (393); of Zebra's 3,278 stripes, 76.9% were CTCF-occupied (2,523), 7.38% were EP-stripes (242), and 15.6% were indeterminate (513); from Stripenn's 873 stripes, 84.5% were CTCF-occupied (738), 2.5% were EP-stripes (22), and 12.9% were indeterminate (113) (Fig. 4A, Supplementary Fig. 11-12). Quagga and Zebra each found 4 to 5 times more stripes than Stripenn, respectively. Quagga's number of called stripes on Hi-C data was the most numerous, and by percent the highest enriched in CTCF. Quagga has the lowest proportion of indeterminate calls, which, from our observation, are highly related to regions with low mappability or bad matrix normalization (Supplementary Fig. 11).

Classifiable stripe calls fall into either dynamic chromatin looping capture or EP interactions; therefore, we assess the quality of each stripe calling tool's outputs by examining if their lengths are reasonable, if CTCF-related stripes are enriched in CTCF/RAD21, and if the stripe calls are generally

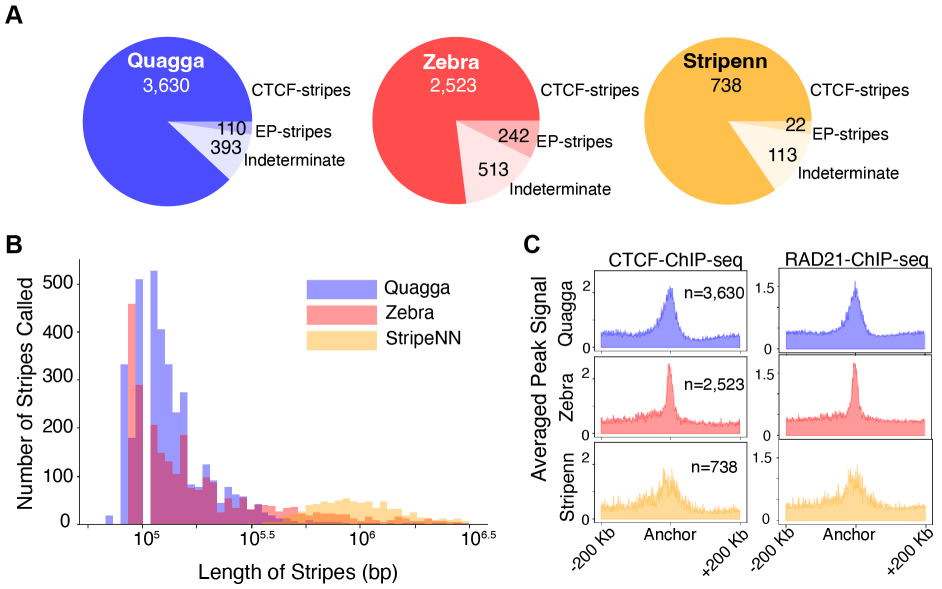

Figure 4: Quagga, Stripenn, and Zebra were compared for their ability to reasonably call CTCF-anchored stripe based on stripe length, CTCF/RAD21 enrichment, and their proportion of off-target stripes found. (A) Distribution of CTCF-stripe lengths from stripes detected by Quagga, Zebra, and Stripenn. (B) Histogram depicting the frequency of stripe occurrence depending on the length of the found stripe for CTCF-anchored stripes. (C) Breakdown of our analysis for stripes belonging to CTCF-stripes, to stripes intersecting enhancer-promoter interacting annotated regions but not CTCF-occupied motifs (EP-stripes), and stripes that do not intersect with CTCF motifs nor EP annotated regions (indeterminate-stripes). Indeterminate stripes may be false positives. Quagga best minimizes indeterminate stripes.

classifiable. To understand if a stripe length is reasonable, we consider that the median size for chromatin looping is 300 kb, where going beyond the median TAD boundary span of 500 kb would suggest a false positive (Hsieh et al., 2022). Quagga and Zebra's distributions of CTCF-occupied stripes align strongly with the expected spans, whereas Stripenn's CTCF-stripes trend longer than the median TAD span (Fig. 4B). Stripenn's trend towards longer stripes may suggest it confuses TAD bounds or compartments for stripes.

CTCF-stripes called by all methods are enriched in CTCF and RAD21, where Quagga and Zebra's aggregated peak signal is strongest (Fig. 4C). Stripenn CTCF and RAD21 peaks are more dispersed and broader. We observe a similar concentrated pileup of CTCF and RAD21 of CTCF-stripes in our benchmarking experiments with downsampled contact maps (Supplementary Fig. 3B) and contact maps of biological replicates (Supplementary Fig. 5B-C). This indicates that Quagga's detection of CTCF-stripes is consistent across different conditions. The CTCF/RAD21 concentration is visible across 5 kb or 10 kb Hi-C resolution, and we observed higher enrichment of CTCF and active histone

marks at 5 kb-detected stripes (Supplementary Fig. 4C), which may reflect a more accurate localization of stripe anchors at higher resolution.

Quagga EP-stripes have strong histone modification signal related to enhancer/promoter activity

Similarly to CTCF-stripes, we report each tool's capability for capturing enhancer-promoter or promoter-promoter interactions, which for simplicity are termed EP-stripes. We designate EP-stripes as those stripes that, along their narrow width, are deficient in CTCF-occupied CTCF motifs and that are intersecting with any number of a set of ChromHMM annotations for GM12878 related to EP-interactions. From all tools' EP-interacting stripes, we demonstrate that Quagga's and Zebra's EP-stripes' median lengths are well below CTCF and TAD median spans (Fig. 5A), and near enough to the median span for EP-interactions (41 kb for enhancer-promoter and 71 kb for promoter-promoter interactions) (Hsieh et al., 2022). Quagga's distribution of lengths demonstrates the bulk of Quagga and Zebra's calls between this reasonable span of lengths, while Stripenn's called stripes are somewhat longer than what is the expected median for EP-interacting stripes. We also demonstrate a variety of widths we solve in Quagga, with narrower stripes solved in comparison to Stripenn (Fig. 5B).

To assess EP-interacting stripes, we investigate the pileup of certain histone modifications around their main anchor position. Starting at the center of the width, we average the window of 200 kb upstream and downstream of the anchor position across all stripes in the designated area. We examine H3K4me2 (strongly associated with promoters and enhancers), H3K4me3 (strong correlation to promoters), and H3K27ac (active enhancer mark) (Calo and Wysocka, 2013). We compare CTCF-stripes to EP-stripes to emphasize the difference in signal. Enhancer/Promoter activity forms very strong peaks in EP-stripes but is greatly diminished in CTCF-stripes in Quagga and Zebra (Fig. 5C). Stripenn shows H3K4me2/3 and H3K27ac are absent in CTCF-stripes, but those signals are also absent in its EP-stripes. It is reasonable that Stripenn's signals do not agree with EP-interaction behaviors as the lengths of its stripes trend very long, close to the behavior of TADs or long loops (300 kb to 1 Mb), so Stripenn's EP-stripes are most likely off-target, and not EP-interactions.

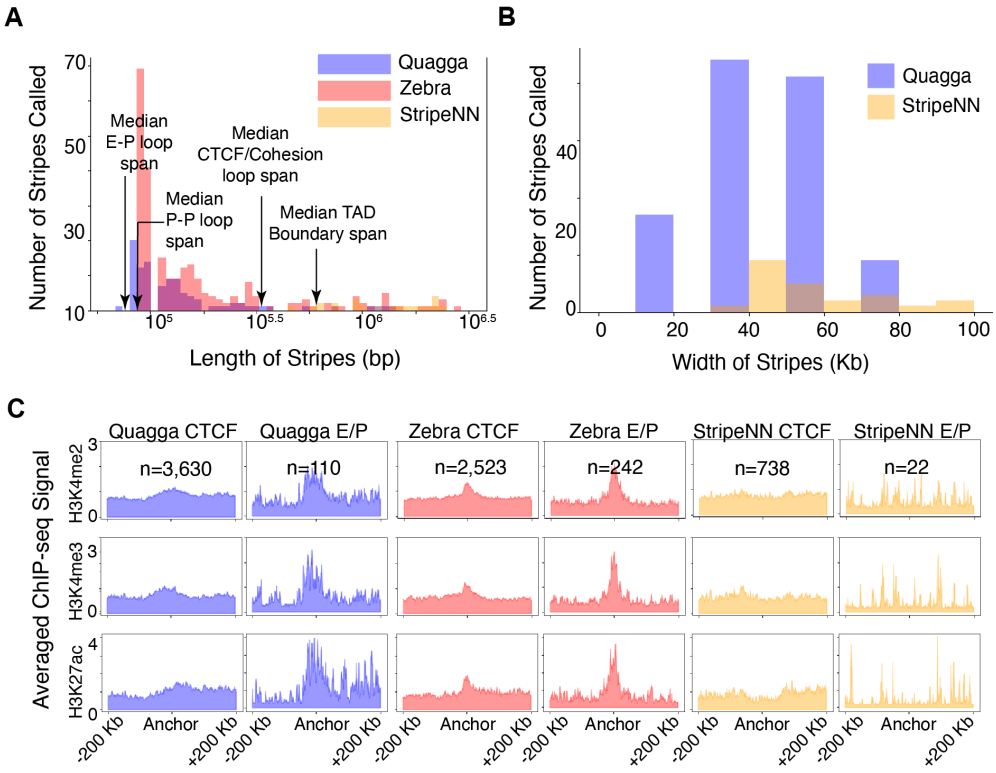

Figure 5: Comparison of stripes called for Quagga, Stripenn, and Zebra were compared for their ability to call enhancer-promoter (EP) interaction-related stripes from GM12878 Hi-C data. The relevant EP-stripes were determined using ChromHMM annotations and unoccupied CTCF motifs. From filtered stripes, their distributions of (A) stripe length and (B) stripe width were determined. (C) Comparison of enrichment of EP-associated ChIP-seq signals between stripe calling tools; CTCF-occupied stripes are compared to EP-stripe s on averaged ChIP-seq pileup over a window of $\pm$200 kb from stripe width's coordinate mid-line for H3K4me2, H3K4me3, and H3K27ac which are associated with EP interactions.

With accurate EP-stripes called by Quagga, we summarized the key characteristics of EP-stripes. First, the length is generally shorter than CTCF-stripes, which is near the median span of enhancer-promoter loops. Second, it is generally weaker than CTCF-stripes, as demonstrated by APA analysis (Supplementary Fig. 12), indicating an alternative mechanism for stripe formation. Just as with our comparison across tools, we observe the same EP-associated histone-tail modifications, showing cleaner peaks at 5 kb over 10 kb (Supplementary Fig. 4C), which in our other experiments, Quagga is sensitive to with a sequencing depth of as low as 125M read-pairs (Supplementary Fig. 3B).

Quagga is effective in Micro-C for H1-hESC EP-stripe detection

We investigate stripe detection on Hi-C and Micro-C for H1-hESCs (H1) to compare Quagga performance on the two data types, and Quagga's ability to call classifiable stripes and minimize indeterminate stripes. Quagga run on H1 Hi-C calls 9,677 stripes at p-val < 0.05: 90.9% CTCF-stripes, 3.97% EP-stripes, and 5.1% indeterminate (Fig. 6A). These percentages are consistent with our analysis of Quagga on GM12878 Hi-C. Quagga on H1 Micro-C calls 4,192 stripes with more CTCF specificity and less indeterminate: 94.6% CTCF-stripes, 3.8% EP-stripes, and 1.6% indeterminate (Fig. 6B). Quagga run on either H1 or GM12878 still has a percentage of indeterminate stripe calls far below Zebra's (15.6%) or Stripenn's (12.9%). Quagga calls a much lower percentage of indeterminate stripes in Micro-C compared with Hi-C, which is related to better qualities of Micro-C data. Quagga's assigned stripe lengths fall within the expected median length for EP/PP interactions (41 and 71 kb, respectively), CTCF-loops (300 kb), and below median TAD spans (500 kb) (Fig. 6A-B). Quagga's stripe calls in H1-hESC Hi-C are nearly double those in H1 Micro-C, perhaps due to differences in distance decay of contacts. For example, H1 HiC has 3.32 billion filtered read pairs with 51.0% being CIS Long (>20 kb), but while H1-Micro-C similarly has 3.4 billion filtered read pairs, they are only 22.9% cis-long; these differences may affect the size of found stripes (Dekker et al., 2017).

Higher Hi-C stripe counts in H1 compared to GM12878 used in the comparison of tools are due to differences in sequencing strategies; the Hi-C protocol for GM12878 used formaldehyde (FA) crosslinking and MboI digestion, whereas H1-hESC Hi-C was crosslinked with both FA and disuc-

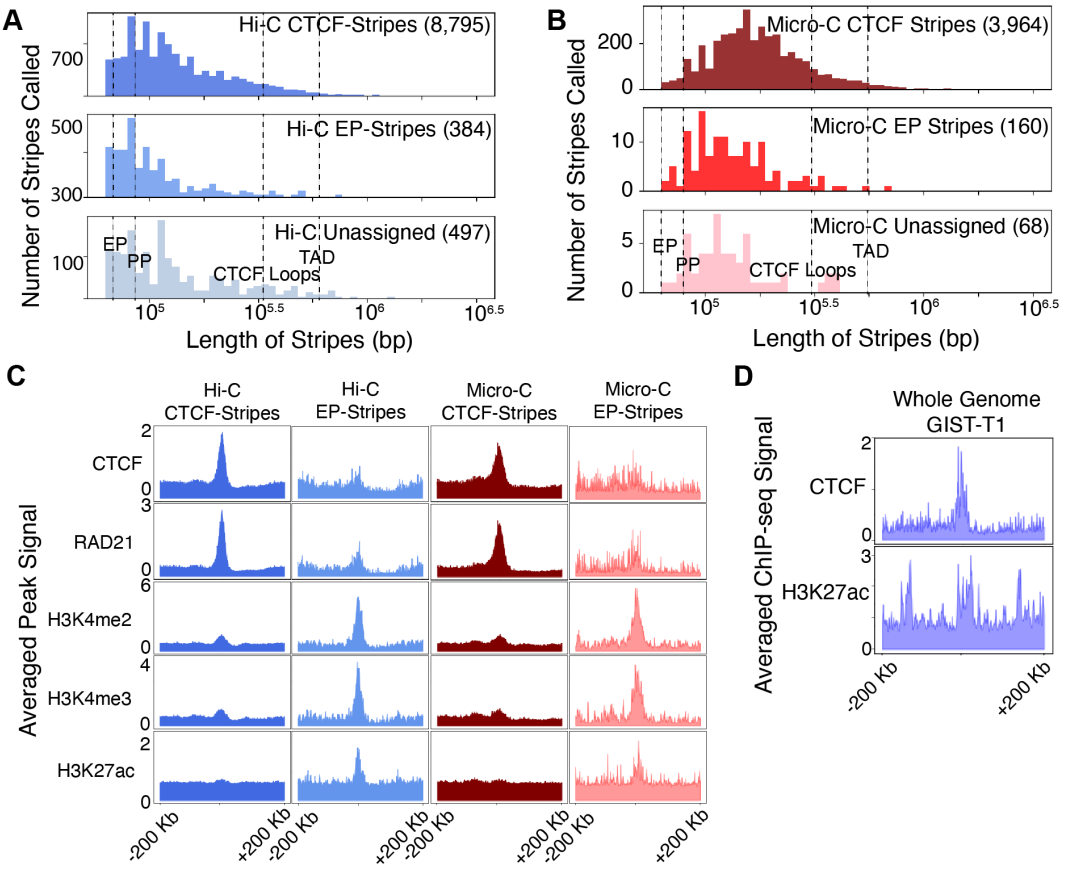

Figure 6: Comparing architectural stripes Quagga finds from Hi-C or Micro-C, their behavior is assessed as CTCF-stripes, EP-stripes, or as indeterminate stripes. (A) the distribution of stripe-lengths on Hi-C data fall into a majority around CTCF-stripe expected lengths. (B) Micro-C is similarly distributed, with less indeterminate stripes compared to indeterminate. (C) In an example stripe in H1-hESC Micro-C, no CTCF signal is present but H3K4me2 is present, suggesting this is an EP-stripe.(D) We show that global stripe detection on region capture Micro-C data recapitulates our observations in Hi-C and Micro-C for gastrointestinal stromal tumour (GIST-T1) cells.

cinimidyl glutarate, digested with DpnII (Rao et al., 2014; Akgol Oksuz et al., 2021); the latter is a recent optimized sequencing protocol that resulted in better signal and loop recall, and it has been known that DpnII digestion is insensitive to CpG methylation where MboI is not (Belaghzal et al., 2017).

The difference in called stripe length and the number of stripes called is supported in the aggregate epigenomics pileup, where we performed an epigenomic pileup of averaged transcription factor signal for CTCF-stripes or EP-stripes for Quagga stripe calls on Hi-C and Micro-C. CTCF-stripes were greatly enriched in CTCF and RAD21 in CTCF-stripes, whereas called EP-stripes were relatively dispersed or lacked well-defined CTCF or RAD21 peaks (Fig. 6C). For histone modifications char-

acteristic to EP-interactions, H3K4me2/3 and H3K27ac, these are more enriched in EP-stripe pileups than with the CTCF-stripes (Fig. 6C). Quagga H1 Hi-C peaks observed in isolating EP-stripes are recapitulated by the proportional percentage in Micro-C, and their epigenomic pileups match expectations for EP-stripe behavior: CTCF and RAD21 depleted, and H3K4me2, H3K4me3, and H3K27ac increased. This trend of enrichment is consistent with our expectation of depleting CTCF-anchored stripes (and, consequently, their loops) and also enriching a list of EP-stripes. Generally, the shorter stripes in H1 Hi-C tended to have clearer CTCF and RAD21 peaks while H3K4me2/3 in H1 Micro-C appear to have marginally stronger epigenomic pileup peaks (Fig. 6C). Micro-C is known for a stronger distance decay factor and has a limited range for reliable signal beyond the main diagonal, so it is understandable that so many fewer CTCF-type stripes may be called. Despite the lower count, CTCF or EP classified stripes show a clear signal in their respective categories. Quagga was also able to find architectural stripes using region capture Micro-C data from gastrointestinal stromal tumour cells (Kim et al., 2024), where we recapitulate the trend of CTCF and enhancer-related H3K27ac aggregation along the long axis anchor of the architectural stripes that Quagga calls there (Fig. 6D).

Discussion

While it is known that architectural stripes appear in bulk sequencing 3C-type experiments like Hi-C, there is still a need for a consistent stripe-calling tool and an adaptable analysis that can be applied rapidly. With Quagga we attempt to handle some of the technical issues we felt other tools struggle with, namely determining stripe length and stripe width accurately, and calling stripes that are enriched in indicators suggestive of CTCF/Cohesin activity or enhancer-promoter interactions. Determining accurate length relies on an accurate stripe start position, a hallmark of Quagga. We also report on an analysis to easily assign stripes as CTCF-stripes or as dynamically interacting EP domains, EP-stripes.

Quagga was able to call chromatin stripes that are notably unique in comparison to Stripenn and Zebra, where we succeeded in mapping most of the chromatin stripes to CTCF-stripes. We also demonstrate the consistency of calling stripes across different resolutions and biological replicates, determining an effective sequencing depth for as low as 250 million filtered read-pairs for Hi-C stripe

detection. Quagga capably calls chromatin stripes in other datasets, particularly Micro-C, HiChIP, and region capture Micro-C.

Consistent with biological expectations of stripes having similar behaviors with a dynamic range of loops or interactions that share a major anchor-point, we demonstrated that chromatin architectural stripes are enriched at their major axis in either CTCF/RAD21 or in histone modifications typically associated with enhancer-promoter activity. Chromatin stripes in Hi-C datasets typically exhibited a range of stripe lengths about the size of median loop spans, which is between the TAD spans and E/P interactions, highly suggestive that the determined CTCF-stripes are neither TADs nor EP interactions. Our results show a clear majority of called stripes in Hi-C are CTCF-stripes, supported by our findings calling architectural stripes on CTCF-degradation assays induced with Auxin, where nearly all chromatin stripes are removed. We also show Quagga detects enhancer-promoter stripes in H1 and GM12878 Hi-C and H1 Micro-C, where the limiting factor is how well the user can assign to such regions; these require existing ChIP-seq datasets and ChromHMM assigned regions, but generally, we found existing ChromHMM assignments to H1 and GM12878 to be effective.

Notably, we demonstrate that the stripes that are not CTCF sensitive have overlap with occupied ChromHMM regions, fitting the profile for Enhancer-Promoter/Promoter-Promoter interactions. The scheme for assigning EP/PP stripe works well in Zebra and Quagga. Quagga provides the capability to record EP-stripes that represent chromatin organizational dynamism at the population level. Not every chromatin stripe calling tool is useful for EP-interaction stripe detection; we show Quagga is capable of removing false positives of the smaller scale that would interfere with the correct detection of EP/PP interacting stripes. The EP/PP assignments fit well within the expectations for CTCF-insensitive EP/PP interactions in terms of H3K4me2 and H3K4me3 expression, with mixed results in H3K27ac; the observed stripe lengths are also consistent with what we expect to see in short-range EP/PP interactions: around the expected span of 41 kb for EP or 71 kb for PP. Notably, EP/PP interactions are not limited to CTCF-insensitive interactions, so these findings are quite conservative.

Stripe detection will continue to be a challenging task, and the need for singular data type prediction algorithms and models will continue as long as data generation schemas are expensive and experimentally challenging. Quagga is an important, rapid, and cost-effective solution for identifying

architectural stripes at the genome scale when lacking ChIP-seq or other sequencing data, where only Hi-C or another 3C family data are available.

Methods

Quagga

Quagga is a chromatin stripe detection tool that works on 3C family sequencing types, particularly Hi-C and Micro-C, but Quagga also works with other contact frequency matrices, including HiChIP. Quagga uses image and signal processing techniques to do naive stripe calling, poisson sampling over the length of stripes to determine stripe length, and scores them with *p*-value (Fig 1A). Quagga functions as a command-line application and a Python library, consisting of a main application but makes its utilities and matrix operations available through the library's Quagga object. Users may take advantage of the included hg38, hg19, mm9, or supply their own. Users may also specify their own parameters due to their needs in stripe detection, such as maximum distance off the main diagonal or how many cores to use. Additional information on installation and usage is documented on the GitHub page https://github.com/dmcbffeng/StripeCaller/.

Input/Output

Quagga requires an HIC or COOL file as well as a standard assembly file that matches the organism/version used to generate the Hi-C file. Quagga returns a BED-like file that specifies the chromosome, coordinates of the stripe, its coordinates reflect the solved width, and the *p*-value determined. Users may specify a cutoff *p*-value for the ending stripes to be considered. Stripe integrity is already considered when forming the *p*-value and the most full/intact stripe is output for any candidate; therefore, an integrity score is not necessary.

Quagga Naive Stripe Calling

1. *Horizontal or Vertical Matrix loading and Filtering*: Before peak finding, Quagga forms a submatrix that reaches between the main diagonal to the max distance specified; Quagga does this horizontally for 5' stripes and vertically for 3' stripes. Depending on the direction of the matrix, we subject it to Gabor filtering, setting the phase to be aligned with the direction of

the stripe: 0 degrees for 5' stripes and 90 degrees for 3'. Gabor filtering clarifies signal in the expected direction of movement, and enhances the signal to noise ratio.

2. *Row/Column Summation and Peak finding*: To solve the matrix peaks we sum the rows of the horizontally loaded submatrix (or columns of the vertically loaded submatrix) to extract a 1-D vector of the filtered Hi-C signal. After applying Gaussian blurring, this vector is subjected to local maxima detection. Quagga uses the "find_peaks()" function from SciPy to solve for maxima along the 1-D signal and produce coordinates that represent the mid-line point of each stripe. The rel_height argument is also incorporated to determine the stripe's width (Fig 1B). The coordinate and width are essential in determining if the called stripe is statically relevant.

Statistical tests of stripe-like patterns

All stripes called are passed through a statistical test to determine if it is an artifact related to noise, and to determine the starting and ending positions of the stripe relative to the main diagonal.

For example, to identify horizontal stripes, Quagga first calculates the observed contact value of each pixel on a candidate stripe and its neighbor regions. The observed value of the interaction between bins i and j (i ¡ j) is

$$\text{Obs}_{i,j} = \text{median}_{j-w<k<j+w} M_{i,k} \tag{1}$$

in which M represents the normalized contact matrix and w is the window size. The use of the median, as opposed to the mean, mitigates the influence of significantly large values resulting from chromatin loops. The average of neighboring regions is then calculated as follows:

$$\text{Exp}_{(i,j)} = \max(\text{median}_{i-w<l<i,j-w<k<j+w} M_{l,k}, \text{median}_{i<l<i+w,j-w<k<j+w} M_{l,k}) \tag{2}$$

in which the top and bottom neighbor regions are calculated separately. Taking the maximum of the two neighboring regions avoids TAD boundaries being called. To assess whether a pixel exhibits significantly higher contacts than its upper/lower neighbor regions, we employ Poisson statistics to derive a corresponding *p*-value.

Calculating stripe lengths

Due to sparsity or noise, the enrichment might not be significant for some pixels along the stripe. Therefore, after obtaining all *p*-values along the horizontal line, we allow breaking points when identifying stripes. The start and end positions are pinpointed as follows.

$$head, tail = \text{argmax}_{st,ed} \sum_{st}^{ed} \log \frac{thr}{P_i} \tag{3}$$

in which $head$, $tail$ are two ends of the stripe, thr is the threshold and P_i is the *p*-value of pixel i. This maximization is performed with the efficient dynamic programming approach (Fig 1C, bottom).

In the end, the significance (*p*-value) of the stripe is determined through the calculation

$$P_{stripe-i} = \exp(\text{mean}_{head \leq k \leq tail}(\log P_{i,k})) \tag{4}$$

Minimizing computational cost

During Quagga's statistical tests, the slowest step is calculating the Poisson *p*-values. Since the significance of every pixel along the candidate stripe needs to be evaluated, Poisson *p*-values are calculated (number of candidate stripes) $\times$ (maximum range for calculation) / (resolution) times, which is usually more than 1 million. We accelerate Quagga's calculation by storing all *p*-value calculation results in a self-balancing tree. Quagga only calculates *p*-values from scratch when the expected/observed values are already stored.

Let $P(E, O)$ denote the *p*-value of Poisson statistics if the expected value is E and the observed value is O. Since the Poisson distribution is discrete, $P(E_i, O_i) = P(E_i, \text{floor}(O_j)))$. Quagga stores all calculated floor(O_j) integer values. For each floor(O_j) value, it stores all calculated E_i in an AVL tree, which ensures the values are well sorted and can be queried and inserted at a short $O(logN)$ time. When calculating $P(E_j, O_j)$, Quagga searches the AVL tree with observed value = floor(O_j) and checks whether a similar expected value E_i was calculated before for this observed value. If an E_j with $\frac{|E_j - E_j|}{E_i} < 1\%$ is found, we directly adapt the pre-calculated result of $P(E_i, \text{floor}(O_j))$ as $P(E_j, \text{floor}(O_j))$. In our benchmarking experiments, this approach accelerates Quagga's calculation by 100 times and only introduces $< 1\%$ error in *p*-value estimation.

Datasets

To establish the performance of all the tools against one another, and to determine the biological activity and validity of the computationally found stripes, we use many publicly available datasets. We list these in their entirety in Supplementary Tables 2 and 3; each following subsection will describe the cell line and data type in each use case.

We build biological replicates of GM12878 Hi-C with data from 4DN Data Portal. In order to get a balanced sequencing depth of two replicates, we merge their technical replicates 1, 2, 3, 4, 5, and 11 of biological replicate 1 to obtain "biological replicate 1" in our study, and merge their biological replicates 3, 4, and 5 to obtain "biological replicate 2" in our study. Both replicates have $\sim$1,100 filtered reads.

Comparison of Stripe Calling Methods and agreement of stripe calls

We explore the differences in tools and briefly calibrated stripe calls by choosing several regions of GM12878 hg38 Hi-C data and testing the ability of Quagga, Stripenn, Chromosight, and Zebra to find the hand-identified stripes, termed "diagnostic stripes". Quagga is available through our GitHub at https://github.com/dmcbffeng/StripeCaller. The chromatin stripe caller, Zebra, is an implementation based on the original paper that programmatically identifies chromatin stripes, accessible via GitHub: https://github.com/XiaoTaoWang/StripeCaller. The tools Stripenn (Yoon et al., 2022) and Chromosight (Matthey-Doret et al., 2020) were used as-is from their respective publications.

A reasonable set of parameters were established and run on hg38 GM12878 Hi-C dataset for all tools, and the recorded stripes were compared, for the diagnostic stripe and exclusion of obvious false positives such as coordinates on the contact frequency matrix that neighbor zero-signal regions, and the start and end regions of a chromosome. By comparing the particular behaviors of each tool on the diagnostic stripes, as well as the off-target calls within those regions, an assessment of how each stripe caller is different can be made.

We used the default parameters for baseline methods, with only two modifications to avoid calling too many stripes: for Chromosight p = 0.3, and for Zebra fold_change=1.4.

An interval intersection algorithm was written to detect consensus stripes between Zebra, Stripenn, Chromosight, and Quagga. Stripe coordinates of the base positions X1, X2 that are used in defining stripe width are used as the interval bounds. An intersection over the left (X1) or right (X2) bounds, or containment of either stripe by the other constituted detection as a matching, similar stripe. An allowance of ± 2 kb was given to account for stripe caller error due to biological noise. Matplotlib Venn diagram's package was used to visualize the resultant stripes, shown in Supplemental Fig. 7.

Aggregated Peak Analysis of Stripe Lengths and Widths, and their Distributions

To pile up the peaks to study stripe widths and their context sequences, we capture all 2 Mb windows around the main diagonal of the chromatin contact matrix that intersects with each stripe's major axis, averaging them for each tool: Zebra, Quagga, and Stripenn. Here, we still use GM12878 Hi-C data.

To study the relative integrity and distribution of lengths, we performed a size-scaling by lengths. For each stripe, we select the Hi-C region centered with width = 100 kb and length = 2 $\times$ length called by the stripe caller, and then zoom in/out to a fixed matrix size. For 3' stripes, we rotate the region by 90 degrees so that the stripe region can overlap with 5' stripes in the pile-up analysis.

The distributions for length and width of stripes were done, when possible, by determining the long and short "sides" of the rectangle each stripe caller solves, deeming the long side the length and the short side the width. Zebra may call multiple stripes along the same length, so we merged Zebra's many calls along a length when determining the 2 Mb APA window or in other pileups, to avoid multiple counting. All visualizations were done with Matplotlib.

CTCF and RAD21 Enrichment Analysis

ChIP-seq data of GM12878 CTCF and RAD21 are used to do ChIP-seq 1D pileup analysis. For pile-up analysis in Fig. 3A-B, we first select the stripe anchor and its ± 1.5 Mb region (i.e., ± 301 bins at 10 kb resolution) for each stripe, and then count the average number of CTCF/RAD21 binding sites in each bin. We used CTCF binding motifs annotated by MotifMap to determine the orientation of CTCF binding. CTCF binding sites without CTCF motifs or with motifs in both orientations are

removed in pile-up analysis in Fig. 3C-D.

Stripe Identity Assignment

Based on the intersection of our stripe's narrow side (the stripe's width) we determine the stripe's identity as CTCF/Cohesin-based stripes, enhancer-promoter interacting stripes, and indeterminate stripes. Intersections are found using a variation of a classical interval intersection algorithm, with a spacer to reflect inclusion or exclusion regions.

CTCF-stripes

To assign which stripes are CTCF/cohesin-loop based stripes, we intersect our stripes with JASPAR CTCF-motifs and check their occupancy state using GM12878 CTCF ChIP-seq peaks (Supplementary Table 2); stripes whose main axis anchor intersects with an occupied-CTCF motif were designated to be stripes composed of dynamic CTCF/cohesin loops, which we called CTCF-stripes. A spacer of $\pm$50 kb along the coordinates along the stripe's width was included in the expanding intersection. Unoccupied CTCF motifs are not included for this assignment.

Enhancer-promoter interacting stripes

Following previously established convention, we group all enhancer-promoter and promoter-promoter activity together as enhancer-promoter interactions to describe EP-stripes. To determine enhancer-promoter (EP) interacting stripes (EP-stripes), we first determine what stripes have no CTCF motif or unoccupied-CTCF motifs (non-CTCF), found using a spacer exclusion-region of $\pm$50 kb along the stripe's width. EP-stripes were determined by intersecting CTCF-deficient stripes with EP regions annotated by established 15-state ChromHMM annotations to identify enhancer/promoter regions (Ernst and Kellis, 2012; Roadmap Epigenomics Consortium et al., 2015). We simplify the state annotation by combining all promoter-promoter and enhancer-promoter activity (1:TssA, 2:TssAFlnk, 3:TxFlnk, 6:EnhG, 7:Enh, 11:BivFlnk, 12:EnhBiv) as other studies have, into an umbrella of "enhancer-promoter" interaction (Roadmap Epigenomics Consortium et al., 2015; Hsieh et al., 2022). Intersections of non-CTCF stripes with EP annotations use an inclusive spacer of $\pm$2 kb.

Indeterminate stripes

The subset of non-CTCF stripes that are not assigned to enhancer promoter activity is deemed "indeterminate" stripes. These stripes intersect with neither occupied CTCF motifs nor with ChromHMM state annotations linked with enhancer/promoter activity.

Epigenomic pileup analysis

As hundreds or thousands of stripes may be called, we sought to analyze our findings at a mass scale, so the aggregate epigenomic pileup analysis visualizes context sequence at and around called stripes to determine on-target/off-target calls. ChIP-seq for H3K4me1, H3K4me2, H3K27ac, CTCF, and RAD21 were used. For each pileup, we centered a window on the coordinate midpoint of the called stripe along its width; In the case of studying Hi-C and Micro-C we used $\pm$200 kb windows, averaging the resultant vector by stripe count. In the example Micro-C stripe, a window size of $\pm$1.5 Mb was used. 500 nt bin ChIP-seq peaks were used in the pileups for all figures.

Comparison of H1-hESC Hi-C versus Micro-C stripes

We repeated our procedure used in GM12878 Hi-C for H1-hESC for Hi-C and Micro-C datasets. A similar strategy was used with H1-hESC CTCF motifs to determine CTCF-stripes and non-CTCF stripes. We then used hg38 liftOver H1 ChromHMM annotations from another study, intersecting enhancer/promoter regions with stripes with no CTCF motifs or unoccupied CTCF motifs and designating these as likely EP-stripes (Ernst and Kellis, 2012; Roadmap Epigenomics Consortium et al., 2015).

Software availability

The Quagga Python package is available on GitHub at https://github.com/dmcbffeng/StripeCaller and in the Supplementary Code. The code used to generate the figures in this manuscript is available at https://github.com/seanpatrickmoran/Quagga_Analysis and in the Supplementary Code.

Competing interest statement

The authors declare no competing interests.

Acknowledgments

The study was supported by the National Human Genome Research Institute (R35HG011279) and by the National Heart, Lung, and Blood Institute (R01HL170115).

Author Contributions: Conception: S.M. and F.F. conceived of and developed the software and method "Quagga". S.M. and F.F. analyzed and interpreted the final results. S.M. and F.F. wrote the manuscript with input from A.H., X.Z., and J.L. All authors have read and approved the final version of the manuscript.

References

B. Akgol Oksuz, L. Yang, S. Abraham, S. V. Venev, N. Krietenstein, K. M. Parsi, H. Ozadam, M. E. Oomen, A. Nand, H. Mao, R. M. J. Genga, R. Maehr, O. J. Rando, L. A. Mirny, J. H. Gibcus, and J. Dekker. Systematic evaluation of chromosome conformation capture assays. *Nat. Methods*, 18 (9):1046–1055, Sept. 2021.

E. J. Banigan, A. A. van den Berg, H. B. Brandão, J. F. Marko, and L. A. Mirny. Chromosome organization by one-sided and two-sided loop extrusion. *Elife*, 9, Apr. 2020.

C. Barrington, D. Georgopoulou, D. Pezic, W. Varsally, J. Herrero, and S. Hadjur. Enhancer accessibility and CTCF occupancy underlie asymmetric TAD architecture and cell type specific genome topology. *Nature Communications*, 10(1):1–14, 2019.

H. Belaghzal, J. Dekker, and J. Gibcus. Hi-C 2.0: An optimized Hi-C procedure for high-resolution genome-wide mapping of chromosome conformation. *Methods*, 123:56–65, 2017.

E. Calo and J. Wysocka. Modification of enhancer chromatin: what, how, and why? *Mol. Cell*, 49 (5):825–837, Mar. 2013.

I. F. Davidson, B. Bauer, D. Goetz, W. Tang, G. Wutz, and J. M. Peters. DNA loop extrusion by human cohesin. *Science*, 366(6471):1338–1345, Dec 2019.

J. Dekker, A. S. Belmont, M. Guttman, V. O. Leshyk, J. T. Lis, S. Lomvardas, L. A. Mirny, C. C. O'Shea, P. J. Park, B. Ren, J. C. R. Politz, J. Shendure, S. Zhong, and the 4D Nucleome Network. The 4D nucleome project. *Nature*, 549:219–226, 2017.

J. R. Dixon, S. Selvaraj, F. Yue, A. Kim, Y. Li, Y. Shen, M. Hu, J. S. Liu, and B. Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485 (7398):376–380, 2012.

J. Ernst and M. Kellis. ChromHMM: automating chromatin-state discovery and characterization. *Nat. Methods*, 9(3):215–216, Feb. 2012.

G. Fudenberg, M. Imakaev, C. Lu, A. Goloborodko, N. Abdennur, and L. A. Mirny. Formation of chromosomal domains by loop extrusion. *Cell Rep.*, 15(9):2038–2049, May 2016.

M. Gabriele, H. B. Brandão, S. Grosse-Holz, A. Jha, G. M. Dailey, C. Cattoglio, T.-H. S. Hsieh, L. Mirny, C. Zechner, and A. S. Hansen. Dynamics of CTCF- and cohesin-mediated chromatin looping revealed by live-cell imaging. *Science*, 376(6592):496–501, Apr. 2022.

G. H. Granlund. In search of a general picture processing operator. *Computer Graphics and Image Processing*, 8:155–173, 1978. URL `https://api.semanticscholar.org/CorpusID:15757543`.

K. Gupta, G. Wang, S. Zhang, X. Gao, R. Zheng, Y. Zhang, Q. Meng, L. Zhang, Q. Cao, and K. Chen. Stripediff: model-based algorithm for differential analysis of chromatin stripe. *Science Advances*, 8(49):eabk2246, 2022.

T.-H. S. Hsieh, C. Cattoglio, E. Slobodyanyuk, A. S. Hansen, O. J. Rando, R. Tjian, and X. Darzacq. Resolving the 3D landscape of transcription-linked mammalian chromatin folding. *Molecular Cell*, 78(3):539–553, 2020.

581 T.-H. S. Hsieh, C. Cattoglio, E. Slobodyanyuk, A. S. Hansen, X. Darzacq, and R. Tjian. Enhancer-
582 promoter interactions and transcription are largely maintained upon acute loss of CTCF, cohesin,
583 WAPL or YY1. *Nat. Genet.*, 54(12):1919–1932, Dec. 2022.

584 K. L. Kim, G. J. Rahme, V. Y. Goel, C. A. El Farran, A. S. Hansen, and B. E. Bernstein. Dissection
585 of a CTCF topological boundary uncovers principles of enhancer-oncogene regulation. *Mol. Cell*,
586 84(7):1365–1376.e7, Apr. 2024.

587 K. Kraft, A. Magg, V. Heinrich, C. Riemenschneider, R. Schöpflin, J. Markowski, D. M. Ibrahim,
588 R. Acuna-Hidalgo, A. Despang, G. Andrey, et al. Serial genomic inversions induce tissue-specific
589 architectural stripes, gene misexpression and congenital malformations. *Nature Cell Biology*, 21
590 (3):305–310, 2019.

591 E. Lieberman-Aiden, N. L. van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit,
592 B. R. Lajoie, P. J. Sabo, M. O. Dorschner, R. Sandstrom, B. Bernstein, M. A. Bender, M. Groudine,
593 A. Gnirke, J. Stamatoyannopoulos, L. A. Mirny, E. S. Lander, and J. Dekker. Comprehensive
594 mapping of long-range interactions reveals folding principles of the human genome. *Science*, 326
595 (5950):289–293, 2009.

596 C. Matthey-Doret, L. Baudry, A. Breuer, R. Montagne, N. Guiglielmoni, V. Scolari, E. Jean,
597 A. Campeas, P. H. Chanut, E. Oriol, A. Méot, L. Politis, A. Vigouroux, P. Moreau, R. Koszul,
598 and A. Cournac. Computer vision for pattern detection in chromosome contact maps. *Nature*
599 *Communications*, 11(1):5795, Nov 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-19562-7.
600 URL `https://doi.org/10.1038/s41467-020-19562-7`.

601 E. P. Nora, B. R. Lajoie, E. G. Schulz, L. Giorgetti, I. Okamoto, N. Servant, T. Piolot, N. L. van
602 Berkum, J. Meisig, J. Sedat, J. Gribnau, E. Barillot, N. Bluthgen, J. Dekker, and E. Heard. Spatial
603 partitioning of the regulatory landscape of the X-inactivation centre. *Nature*, 485(7398):381–385,
604 2012.

605 S. S. P. Rao, M. H. Huntley, N. Durand, C. Neva, E. K. Stamenova, I. D. Bochkov, J. T. Robinson,
606 A. L. Sanborn, I. Machol, A. D. Omer, E. S. Lander, and E. L. Aiden. A 3D map of the human

genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 59(7):1665–1680, 2014.

Roadmap Epigenomics Consortium, A. Kundaje, W. Meuleman, J. Ernst, M. Bilenky, A. Yen, A. Heravi-Moussavi, P. Kheradpour, Z. Zhang, J. Wang, M. J. Ziller, V. Amin, J. W. Whitaker, M. D. Schultz, L. D. Ward, A. Sarkar, G. Quon, R. S. Sandstrom, M. L. Eaton, Y.-C. Wu, A. R. Pfenning, X. Wang, M. Claussnitzer, Y. Liu, C. Coarfa, R. A. Harris, N. Shoresh, C. B. Epstein, E. Gjoneska, D. Leung, W. Xie, R. D. Hawkins, R. Lister, C. Hong, P. Gascard, A. J. Mungall, R. Moore, E. Chuah, A. Tam, T. K. Canfield, R. S. Hansen, R. Kaul, P. J. Sabo, M. S. Bansal, A. Carles, J. R. Dixon, K.-H. Farh, S. Feizi, R. Karlic, A.-R. Kim, A. Kulkarni, D. Li, R. Lowdon, G. Elliott, T. R. Mercer, S. J. Neph, V. Onuchic, P. Polak, N. Rajagopal, P. Ray, R. C. Sallari, K. T. Siebenthall, N. A. Sinnott-Armstrong, M. Stevens, R. E. Thurman, J. Wu, B. Zhang, X. Zhou, A. E. Beaudet, L. A. Boyer, P. L. De Jager, P. J. Farnham, S. J. Fisher, D. Haussler, S. J. M. Jones, W. Li, M. A. Marra, M. T. McManus, S. Sunyaev, J. A. Thomson, T. D. Tlsty, L.-H. Tsai, W. Wang, R. A. Waterland, M. Q. Zhang, L. H. Chadwick, B. E. Bernstein, J. F. Costello, J. R. Ecker, M. Hirst, A. Meissner, A. Milosavljevic, B. Ren, J. A. Stamatoyannopoulos, T. Wang, and M. Kellis. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539):317–330, Feb. 2015.

T. Sexton, E. Yaffe, E. Kenigsberg, F. Bantignies, B. Leblanc, M. Hoichman, H. Parrinello, A. Tanay, and G. Cavalli. Three-dimensional folding and functional organization principles of the *Drosophila* genome. *Cell*, 148(3):458–472, 2012.

L. Vian, A. Pekowska, S. S. Rao, K.-R. Kieffer-Kwon, S. Jung, L. Baranello, S.-C. Huang, L. El Khattabi, M. Dose, N. Pruett, et al. The energetics and physiological impact of cohesin extrusion. *Cell*, 173(5):1165–1178, 2018.

W. Xi and M. A. Beer. Loop competition and extrusion model predicts CTCF interaction specificity. *Nat. Commun.*, 12(1):1046, Feb. 2021.

S. Yoon, A. Chandra, and G. Vahedi. Stripenn detects architectural stripes from chromatin conformation data using computer vision. *Nature Communications*, 13(1):1602, Mar 2022.

633 ISSN 2041-1723. doi: 10.1038/s41467-022-29258-9. URL https://doi.org/10.1038/
634 s41467-022-29258-9.

635 H. Zhang, D. J. Emerson, T. G. Gilgenast, K. R. Titus, Y. Lan, P. Huang, D. Zhang, H. Wang, C. A.
636 Keller, B. Giardine, et al. Chromatin structure dynamics during the mitosis-to-G1 phase transition.
637 *Nature*, 576(7785):158–162, 2019.