2021 IEEE 3rd International Conference on BioInspired Processing (BIP) | 978-1-6654-2722-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIP53678.2021.9612782

Automated Detection and Localization of Genome Inversions using Principal Component Analysis

Fabian Fallas-Moya
EECS Department
The University of Tennessee
Knoxville, US
ffallasm@vols.utk.edu

Ronald J. Nowling
EECS Department
Milwaukee School of Engineering
Milwaukee, US
rnowling@alumni.nd.edu

Scott Emrich
EECS Department
The University of Tennessee
Knoxville, US
semrich@utk.edu

Amir Sadovnik
EECS Department
The University of Tennessee
Knoxville, US
asadovnik@utk.edu

***Abstract*—Inversions occur when sections of a chromosome (DNA molecule) are completely reversed end-to-end. Large inversions (multiple megabases in length) can be detected, localized, and genotyped using principal component analysis (PCA) of single nucleotide polymorphisms (SNPs). However, detection and localization tasks are performed and interpreted manually. We propose a novel pipeline for the detection and localization tasks in an automated manner. We compare our results with manual analysis for localization and show that our algorithm can achieve a similarity score of 0.95 on average. For the classification task, we achieve an accuracy of 0.88 as compared to manual classification. Our results suggest that our proposed methods are fast and accurate for these tasks and can be used as tools for detection and localization.**

***Index Terms*—chromosome, inversion, pca**

I. Introduction

Inversions occur when a segment of a chromosome is reversed end-to-end. Since large inversions contribute to the population structure and adaptation of some species, the detection of inversions is an important task in understanding how different species are formed and maintained [1], [2]. Figure 1 shows an example of a small inversion.

Anopheles mosquitoes are the primary vector that transmits the parasites that cause malaria [3]. Malaria presents a significant risk to worldwide health. In 2017 alone, malaria caused $435,000$ deaths [4]. Analysis of inversions in these mosquito species is an important step in the process of determining which populations carry insecticide-resistance mutations and to help predict the spread of insecticide resistance. Most known inversions in *An. gambiae* occur on chromosome 2 and may have many inversions (see Figure 2).

There are three distinct inversion analysis tasks: detection (determine if there is an inversion in the data), localization (indicate the boundaries of inversions), and, determining the combinations of orientations that form the genotype of the sample (mosquitoes are diploid, meaning that they have two copies of each chromosome, each of which will have its own orientation of the inversion). We focus on the first two tasks.

Because inversions suppress recombination during meiosis, variants (Single Nucleotide Polymorphisms or SNPs) that occur on one orientation of an inversion are not shared with the other orientation. Correlation of the SNP genotypes with the inversion genotypes allows inversions to be detected

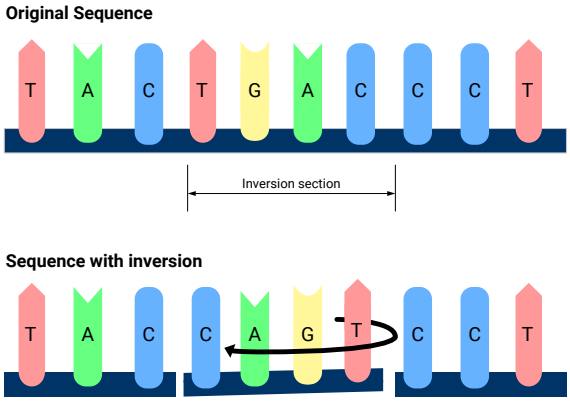


Fig. 1: A visual explanation of an inversion.

through Principal Component Analysis (PCA) of the SNPs. Traditionally, plots are created for a human expert to visually indicate the location of the inversion (if there is one).

The number of SNPs in a dataset can easily number in the millions, making it important to choose computationally-efficient approaches. For example, one of our datasets has 81 mosquito samples with 45 million SNPs of position. One way to visualize the location of potential inversion is by testing each SNP with respect to each PCA component that captures an inversion genotype [6]. Testing the SNPs for association with each principal component requires evaluating 45 million separate statistical tests. When the p-values of the SNPs are plotted along a chromosome, the inverted sections "stand out due to the presence of a step-function-like pattern" [7].

Association Test (AT) based on Logistic Regression has proved to be useful when there is missing data [6]. The models were fit with stochastic gradient descent, which is computationally inefficient for low dimensional data. We propose (section II-A) and demonstrate that a standard statistical test works equally well, allowing the use of substantially faster implementations for association testing. In the next step, we propose a new method (section II-B) that automates the location of potential inversions, which both reduces human error and offers a significant improvement in accuracy and usability over previous methods. Finally, we propose a

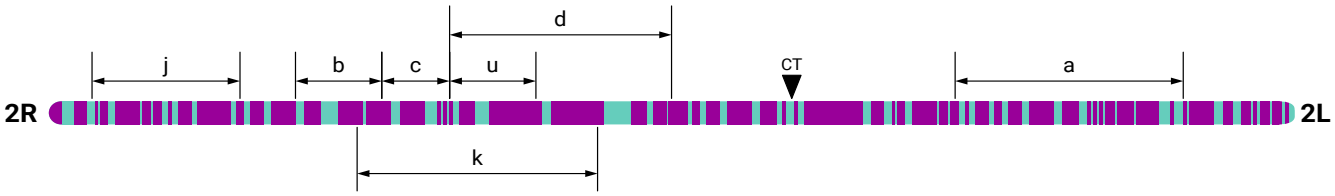


Fig. 2: Approximate diagrammatic representation of common polymorphic inversions on chromosome 2 of *An. gambiae*. The left and right boundaries of the inversions are represented by brackets. A single chromosome may have many different inversions that can complicate analysis. This specific diagram is based on inversions described in [5].

method (section II-C) that determines if a principal component represents an inversion or not. All our code is available at: https://github.com/nowling-lab/asaph.

A. Previous Work

Ma and Amos [8] use PCA to detect and characterize inversions based on SNPs. A sliding window was applied to SNP data to uncover a "three stripe" pattern and reveal inverted genomic intervals. These three clusters correspond to homozygous inverted, homozygous standard, and heterozygous segments. The first principal component on the entire variant dataset was sufficient to uncover inversion-associated SNPs [6]; however, they did not explore the location of inversions.

Analysis of Anopheline mosquitoes, however, is complicated by missing data and confounding (e.g., non-inversion related) signals because of their complex evolutionary histories (Fontaine et al. [9]). Nowling et al. [6] (Asaph version 1) used AT to localize inverted and other relevant sections even in the presence of missing data. However, visual inspection needed to be performed by an expert.

Love *et al.* [3] combined the original approach of [8] with prior biological knowledge about mosquito populations and their inversions. However, in their work, they did not focus on the localization task, which is one of our main concerns.

Finally, inveRsion [10] is an R package that "identifies changes in linkage disequilibrium along the chromosome arm from SNP data to find inversion breakpoints" [7]. Although this work attempts to perform a similar task to ours, we had difficulty running this R package on large files. In addition, their package does not find the exact location of the inversion as our proposed work does.

B. Data

We use single nucleotide polymorphism (SNP) data from two closely related malaria mosquito species available from the Malaria Genomic Epidemiology Network (accessible from https://www.malariagen.net). This resource provides Variant Call Format (VCF) data for individual mosquitoes from multiple African countries encoded as either a 0 (the same allele as a reference mosquito genome) or 1 (alternative allele relative to a reference genome). This VCF data has been previously filtered such that all alleles are biallelic because mosquitoes are diploids (two alleles per individual). A small section is visualized in Figure 3. Figure 2 shows that in a single chromosome we can have one, multiple, or no inversions.

Originally, rows are SNPs (alleles) and columns are individual mosquitoes. We transpose the matrix to apply PCA so that every row is a mosquito sample and every column is an SNP position in the data (see Figure 3). The VCF files we use were filtered to remove variants with low QUAL (quality) scores. Details are given in the supplemental material of Miles et al. [11]. We use data from the *Anopheles* 1000 Genomes consortium [11], the 16 *Anopheles* Genomes [9] consortium, and the *Drosophila* Genetics Reference Panel [12], [13]. Access to all these benchmarks (and datasets) can be found at https://github.com/nowling-lab/asaph/tree/main/inversion-benchmark, where we describe the datasets and cite the references from where we obtained the data. These benchmarks have negatives (no inversions), and single or multiple positives (one or multiple inversions).

II. Methods

Figure 4 shows our proposed approach which has three methods. Our first method uses an alternative statistical test when performing the association tests, which is much more computationally efficient. Our second method performs the potential boundary location of the inversion by using a novel idea of convolution in 1D. Finally, our third method performs density analysis to indicate if the component can be linked to an inversion (or not). These methods are explained in detail in the following sections.

We do not anticipate that polyploidy will be a problem. We primarily use bi-allelic SNPs, meaning that they only have 2 possible alleles at each site. Even if an organism had 4 copies of each chromosome, we could encode the genotype as the number of copies of the reference allele (e.g., 0, 1, 2, or 3). Our correlation-based approach would still be effective in this situation.

A. A Faster Implementation of Association Tests

We first apply PCA to the transposed SNP variant matrix to obtain a matrix of components. As explained in section I, if our SNP variant matrix has a dimensionality of $N \times M$, our matrix of components will have a dimensionality of $N \times 6$ (assuming we want only the first 6 components). Then, we take every component **alone** to statistically compare it against every SNP position. This comparison is known as Association Tests (AT). So, for every component, we end up with a vector of size M. Algorithm 1 shows the pseudo-code of this algorithm (on line 7, we apply our version of association tests).

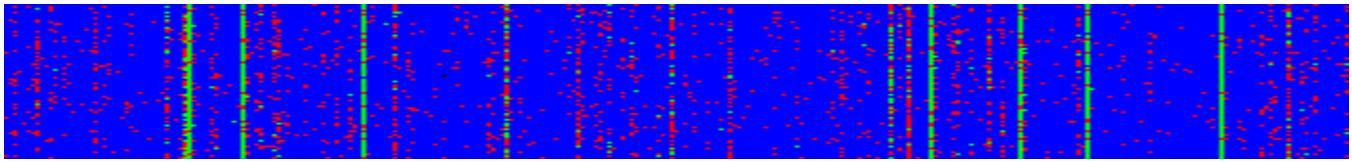


Fig. 3: Visualization of raw SNP data within a subset of the 2La inversion of *An. gambiae*. Color description: 0/1 (one reference and one alternate allele) is red, 1/1 (two alternative alleles) is green, and 0/0 (two reference alleles) is blue. Rows: 81 mosquito samples. Columns: positions in the chromosome that usually are in tens of millions.

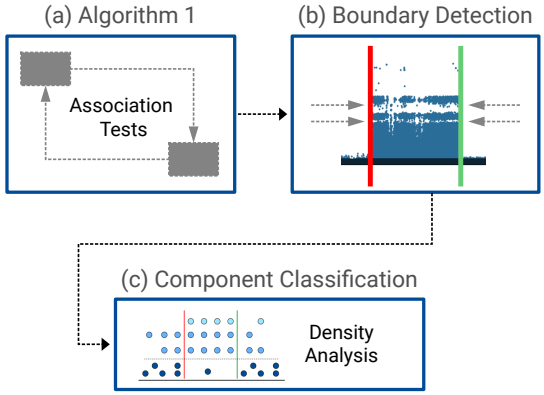


Fig. 4: Our proposed pipeline - with the three methods - for inversion analysis.

Algorithm 1 Performing association tests with a SNP matrix

```
1:  function GET_PVALUES(matrix)
2:      components ← apply_PCA_over_rows(matrix)
3:      pvalue_results ← empty_set
4:
5:      for comp in components do
6:          for snp in matrix do
7:              res_val ← asso_tests(comp, snp)
8:              pvalue_results.append(res_val)
9:          end for
10:
11:         save_results_into_file(pvalue_results, comp)
12:         pvalue_results ← empty_set
13:     end for
14:
15: end function
```

Nowling *et al.* [6] implemented logistic regression (LR) to perform AT. According to Xiao [14], the complexity of LR is given by $O(nd)$, where n is the number of training examples and d is the number of dimensions of the data. If we use multiple components that explain most of the variation in the SNP data, it requires $O(nd \times M)$ where M is the number of SNPs. For example, in our initial 2La experiments the LR-based framework required an execution time of ~ 40 hours.

To lower the complexity, we sought an alternative method that could generate similar results and found that the Pearson Correlation Coefficient (PCC) was a good fit. It has a complexity of O(n) where n is the number of samples. The only modification in the algorithm was to use PCC as the AT function (Algorithm 1, line 7). 2La data requires 0.4 hours on average, which is 17.5X faster than using LR-based association (see results in section III-A).

B. Localization

Prior work has relied on visual inspection of the results to detect boundaries [3], [6]. We propose a two-stage method to locate the potential location of inversions. The first stage finds a horizontal threshold, and the second stage finds the exact vertical boundaries of the inversion. Although our method finds these boundaries, it is possible that the component does not have an inversion, therefore this method finds potential boundaries because the final method of our pipeline (density analysis) will indicate if there is an inversion or not.

1) Horizontal Threshold Detection: Since the p-values are contiguous, a feasible option to find the boundary was to use clustering to clearly define a decision boundary. Our method relies on finding a horizontal threshold using k-means to locate two centroids over the p-values. An important finding is that most of the p-values—more than 95%—are close to 0 (see Figure 5). Based on this observation we apply k-means clustering with only 2 classes, which correspond to either inverted or non-inverted positions. Consequently, the threshold acts like a decision boundary where samples above the threshold are inverted SNPs and the ones that are below it are non-inverted SNPs. The first time that k-means runs, it sets the threshold close to the Y-value of 0.4 (Figure 5) which clearly is not accurate. This happens because of the distance of the cluster centers, so, to solve this inaccuracy of the threshold we apply k-means again over only the densest class, that is, the class that has more SNPs, which happens to be the class that has many p-values close to zero. Now, we have a better threshold. We keep iterating until there are acceptable values in the densities of what is below and above the threshold. Figure 5 shows the final decision boundary with dark-blue and light-blue p-values for every SNP position.

2) Boundary Detection: Figure 5 shows that it is simple to visually determine the approximate breakpoints of the inversion. We want to numerically estimate the location of an inversion, so, we use the horizontal threshold found in the previous section. First, we create windows using all the p-values and use these windows as bins to count the frequency of points that are above the horizontal threshold. The size

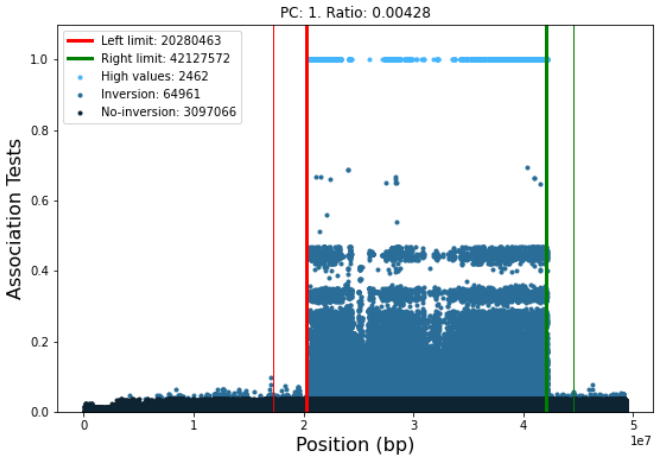


Fig. 5: Results using the first component with 2La. SNPs with no-inversion (low p-values) are represented by the dark-blue points whereas SNPs with inversion (high p-values) are represented by the light-blue and the top-lighter-blue. The threshold is seen as the horizontal boundary between inversion and no-inversion SNPs.

of the windows corresponds to 0.1% of the total number of SNPs. The basic idea is to count p-values that are above the horizontal threshold to have a bin frequency vector.

Then, we utilize an idea inspired by the principle of 1D convolution: using the Sobel operator [15], [16], [17] to detect "edges" in the bin frequency vector. The idea is to travel from left to right to detect (by using a filter with values of the Sobel operator) the leftmost boundary (or breakpoint) of the inversion and then stop when a signal is found. Then, we go from right to left and stop when we detect the rightmost boundary of the inversion. The difference with respect to 1D convolution is that we stop as soon as we find the first "edge". If no edge is found, it means that the given component does not have an inversion. If the two boundaries were found, we still need to perform the next method in the pipeline to know *for sure* if the component has an inversion or not.

C. Classification of Inversion

Finally, we classify the sample using the threshold and the two boundaries/limits (left and right) of the inversion. For this, we considered techniques such as Jensen-Shannon and Kullback-Leibler (KL) divergence but the results were not successful because most samples have p-values close to zero (more than 95%) and these metrics are not intended for such skewed data.

We implement a simple but effective alternative method for classification. By using the potential boundaries of the inversion, we can split our samples into two sets, the set of samples inside the inversion and the set of samples outside the inversion. For each set, separately, we calculate the ratio between the number of samples that are above and below the horizontal threshold, and with these two ratios, we calculate the density ratio. Equation 1 shows the formula.

$$density\_ratio = \frac{out_{above}/out_{below}}{in_{above}/in_{below}} \quad (1)$$

where out_{above} and out_{below} represent points outside of the boundaries (red and green lines in Figure 6), and in_{above} and in_{below} represent points inside the inversion. As seen in Figure 5, the left boundary has two red lines, the section in between these red lines represents a section whose SNPs values were ignored when calculating the density ratio because the borders have outliers in terms of inverted vs. non-inverted, so, by ignoring these small sections we improved our results.

If the *density_ratio* is less or equal than 0.01 then we classify the component as having an inversion and if the result is greater as not having an inversion. We chose the value 0.01 based on all the experiments we performed. Figure 6 shows an example of this method where on the left the *density_ratio* is 0.008 and thus the result is valid for an inversion. On the right, the result is 0.075 which indicates the component does not have an inversion.

III. Experiments and Analysis

In this section, we discuss the efficacy of our (i) faster method for association tests, (ii) boundary detection method, and (iii) classifier of components method.

A. Speedup Calculations

First, we show how our proposed PCC method for the association tests speeds up the execution time with respect to LR. Table I compares the execution time of LR against PCC. These experiments show that our alternative statistical method works 17.5 times faster than using LR. The results of large datasets can be analyzed with less computation and thus, less execution time.

Dataset	Asaph V1 [6] using LR	Our new method - PCC
2La	39.4	2.2
2Rbcd	19.8	1.2
2Rb	32.3	1.8

TABLE I: Running time in hours of three experiments. PCC is substantially faster in all three when compared to LR.

B. Boundary Detection

Love et al. [3] report the localization of inversions for just a few datasets. They used biological tests (experimentally-mapped) to obtain these locations, and we compared their results with our method for localization (see Table II). We use the Jaccard index [18] or Intersection over Union (IoU) to compare the results. This index gives us the similarity between samples by using the size of the intersection divided by the size of the union of the samples.

As we mentioned in the description of the data, we filtered to remove variants with low QUAL (quality) scores. It is true that variants with little support could cause some error in the inversion boundaries. However, the boundary errors are likely

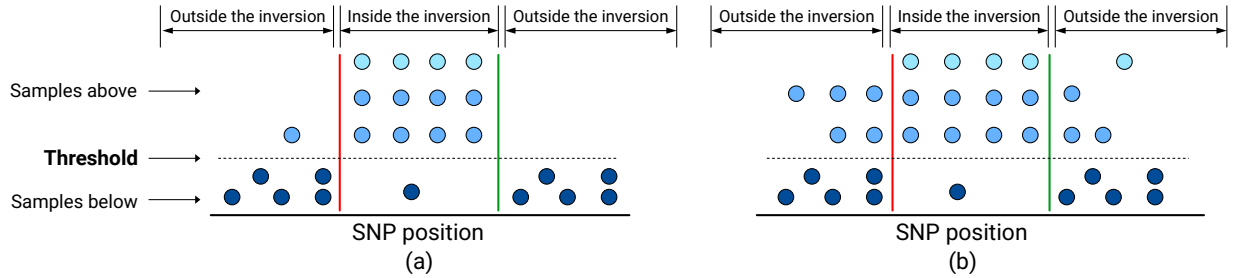


Fig. 6: Density analysis: the ratio of the non-inverted divided by the ratio of the inverted section. (a) Non-inverted is $1/10 = 0.1$, inverted is $12/1 = 12$, so, the density ratio is $0.1/12 = 0.008$. (b) Non-inverted is $9/10 = 0.9$, inverted is $12/1 = 12$, so, the density ratio is $0.9/12 = 0.075$. Density threshold = 0.01, so (a) is classified as an inversion (< 0.01) and (b) is not (> 0.01).

Data	Love et al. [3]	Our method	IoU
2La	20524058-42165532 ([19])	20280463-42127572	0.987
2Rbc	19023925-31473100 ([20], [21])	18994199-31302151	0.984
2Rd	31495381-42375004 ([22])	31310099-42725018	0.953
2Ru	31473000-35505236 ([21])	31414794-36121835	0.856

TABLE II: Results of our boundary detection method for the localization task of the inversion.

to be very small relative to the size of the inversion. However, our results show that the boundaries are similar to the state-of-the-art for these four important inversions in *Anopheles* mosquitoes. As seen, the IoU in every test is high for 2La, 2Rbc, and 2Rd, with an IoU higher than 0.95. 2Ru has an acceptable result with an IoU of about 0.856. In the future, as more biological analysis on inversions becomes available, we will be able to compare more results. Meanwhile, with the available data, our proposed method is therefore effective in finding an approximate location of inversions.

C. Classification of components

We performed 72 runs of experiments to measure the performance of classification. We classify if a specific component has an inversion or not. Figure 7 shows two results of these experiments. We use the 12 datasets described in section I-B and for each dataset, we performed the association tests using its first 6 components from the PCA. We compared the performance of our algorithm vs. an expert classification of the component, i.e., visual inspection of the plot (by an expert).

The precision is 0.94, showing that our method detects inversions when they are actually present. The recall is 0.68, which highlights false-negatives. To increase recall, it can be used a dynamic value (treated as a hyperparameter) for the density analysis vs. the fixed one we are currently using (the fixed value of 0.01), which may help in increasing our recall. That said, our final accuracy is 0.88. This can be considered a high score because the results are compared to manual classification which can be somewhat subjective.

Figure 7 shows the results from the 2Rbcd (2R with b, c, and d inversions) dataset. We show that our methods work as expected. For example, we can see that 7a accurately finds the inversion, and 7b indicates that the component has no inversion. As shown in this Figure, the first component finds the location of the inversion and the classification result also works as expected. Figure 7 shows just two components, but we performed 6 components in every experiment.

IV. Conclusions and Future Work

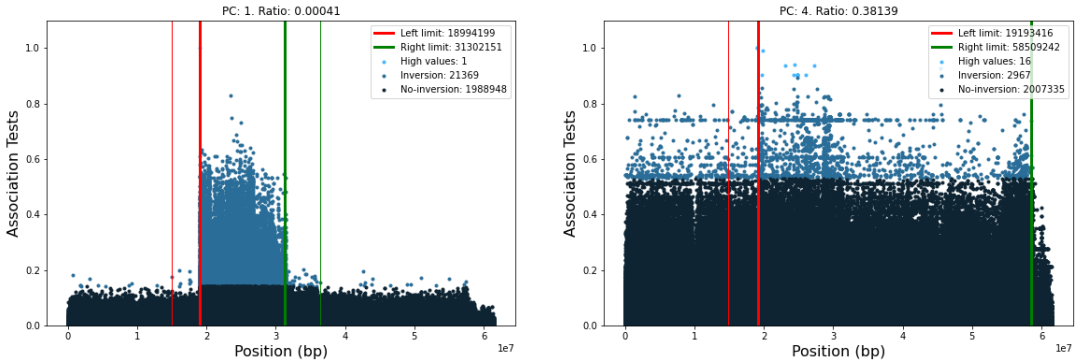
Comprehensive SNP-based inversion detection can be computationally intensive because we have to run a statistical test for every SNP position and only a small number of variants show a high p-value in inverted regions. For example, as shown in Figure 5 only 0.09% of the approximately 3.2 million SNPs are strongly correlated with the inversion. This percentage is even lower for the 2R datasets. This *weak* signal makes this problem difficult for deep learning models that rely on strong features to learn. We show that PCA and association tests are still one of the best options to find inversions. Also, a simple statistical test alternative—correlation—maintains accuracy while being substantially faster than prior methods.

Further, finding the inversion breakpoints, i.e., where the inversion starts and ends on a chromosome, has required labor-intensive manual inspection of raw association data. In this paper, we validate a new precise (quantitative) estimate of these breakpoints, which opens the door for automatic annotation and karyotyping of future genome sequencing samples.

We note that our new framework still relies on the large-scale association of inversions to principal components derived from available SNP data. It will not work well if the available samples are unbalanced and do not have any "stripes" in a PCA ordination. Future work could apply more supervised learning-inspired methods that can predict inversions based on known inverted samples independent of the underlying inversion frequencies in the samples provided for analysis.

We also show that our density analysis provides a tool to classify components with a final accuracy of 0.88 compared to manual classification. This will allow a faster analysis of components with high confidence in the results. To the best of our knowledge, no prior method can classify components at all. An interesting future improvement is related to adjusting the *density_ratio* to improve the performance without benchmarking based on known inversions.

Finally, as future work, it will be interesting to use simulated data that can help us to explore a wide range of impacts on the accuracy of the models. For example, what if you



(a) Successful classification as inversion (density_ratio < 0.01) and localization (green and red lines).

(b) Successful classification as no-inversion (density_ratio > 0.01). Even though, localization works, the density_ratio indicates that there is no-inversion.

Fig. 7: Successful results of two components (first and fourth) over 2Rbcd data.

have fewer SNPs or samples? What if the correlation between the SNPs and inversions are less? What if the areas around the inversion boundaries are noisy? This would allow us to determine how robust the model is. Also, it will be important to also evaluate this method on data from other organisms (other than mosquitoes) to ensure that it generalizes well.

REFERENCES

[1] A. A. Hoffmann and L. H. Rieseberg, "Revisiting the impact of inversions in evolution: From population genetic markers to drivers of adaptive shifts and speciation?," *Annual Review of Ecology, Evolution, and Systematics*, vol. 39, no. 1, pp. 21–42, 2008.

[2] M. Kirkpatrick, "How and why chromosome inversions evolve," *PLOS Biology*, vol. 8, pp. 1–5, 09 2010.

[3] R. Rebecca Love, S. N. Redmond, M. Pombi, B. Caputo, V. Petrarca, A. della Torre, and N. J. Besansky, "In Silico Karyotyping of Chromosomally Polymorphic Malaria Mosquitoes in the Anopheles gambiae Complex," *G3: Genes, Genomes, Genetics*, vol. 9, no. 10, pp. 3249–3262, 2019.

[4] WHO, *WHO — The World malaria report 2018*. 2018.

[5] M. Coluzzi, A. Sabatini, A. Della Torre, M. A. Di Deco, and V. Petrarca, "A polytene chromosome analysis of the Anopheles gambiae species complex," *Science*, vol. 298, no. 5597, pp. 1415–1418, 2002.

[6] R. J. Nowling, K. R. Manke, and S. J. Emrich, "Detecting Inversions with PCA in the Presence of Population Structure," *bioRxiv*, p. 736900, 2019.

[7] R. J. Nowling, K. R. Manke, and S. J. Emrich, "Detecting inversions with pca in the presence of population structure," *PLOS ONE*, vol. 15, pp. 1–20, 10 2020.

[8] J. Ma and C. I. Amos, "Investigation of inversion polymorphisms in the human genome using principal components analysis," *PLoS ONE*, vol. 7, no. 7, 2012.

[9] M. C. Fontaine, J. B. Pease, A. Steele, R. M. Waterhouse, D. E. Neafsey, I. V. Sharakhov, X. Jiang, A. B. Hall, F. Catteruccia, E. Kakani, S. N. Mitchell, Y.-C. Wu, H. A. Smith, R. R. Love, M. K. Lawniczak, M. A. Slotman, S. J. Emrich, M. W. Hahn, and N. J. Besansky, "Extensive introgression in a malaria vector species complex revealed by phylogenomics," *Science*, vol. 347, no. 6217, 2015.

[10] A. Cáceres, S. S. Sindi, B. J. Raphael, M. Cáceres, and J. R. González, "Identification of polymorphic inversions from genotypes," *BMC Bioinformatics*, vol. 13, Feb. 2012.

[11] A. Miles, N. J. Harding, G. Bottà, C. S. Clarkson, T. Antão, K. Kozak, D. R. Schrider, A. D. Kern, S. Redmond, I. Sharakhov, D. Ayala, N. J. Besansky, A. Burt, M. W. Hahn, J. Midega, D. E. Neafsey, J. Pinto, D. Weetman, C. S. Wilding, B. J. White, A. D. Troco, N. Elissa, J. Dinis, C. Mbogo, P. Bejon, J. Stalker, P. Vauterin, B. Jeffery, I. Wright, L. Hart, B. MacInnis, C. Henrichs, R. Giacomantonio, gambiae 1000 Genomes Consortium, D. a. group, P. w. group, S. collections Angola:, B. Faso:, Cameroon:, Gabon:, Guinea:, Guinea-Bissau:, Kenya:, Uganda:, Crosses:, Sequencing, d. production, W. a. development, and P. coordination, "Genetic diversity of the african malaria vector anopheles gambiae," *Nature*, vol. 552, pp. 96–100, Dec 2017.

[12] T. F. C. Mackay, S. Richards, E. A. Stone, A. Barbadilla, J. F. Ayroles, D. Zhu, Y.-Q. Wu, Y. Zhu, C. M. Bergman, K. R. Thornton, D. Mittelman, and R. A. Gibbs, "The drosophila melanogaster genetic reference panel," *Nature*, vol. 482, pp. 173–178, Feb 2012.

[13] W. Huang, A. Massouras, Y. Inoue, J. Peiffer, M. Ramia, A. M. Tarone, L. Turlapati, T. Zichner, D. Zhu, R. F. Lyman, Y. Zhang, Y. Zhu, R. R. H. Anholt, A. Barbadilla, J. S. Johnston, E. A. Stone, S. Richards, B. Deplancke, and T. F. C. Mackay, "Natural variation in genome architecture among 205 drosophila melanogaster genetic reference panel lines," *Genome Research*, vol. 24, pp. 1193–1208, Apr. 2014.

[14] W. Xiao, "An Online Algorithm for Nonparametric Correlations," pp. 1–18, 2017.

[15] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an Image Edge Detection Filter Using the Sobel Operator," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[16] W. Gao, X. Zhang, L. Yang, and H. Liu, "An improved sobel edge detection," in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 5, pp. 67–71, 2010.

[17] N. Kanopoulos, N. Vasanthavada, and R. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[18] S. Fletcher and M. Z. Islam, "Comparing sets of patterns with the jaccard index," *Australasian Journal of Information Systems*, vol. 22, Mar. 2018.

[19] I. V. Sharakhov, B. J. White, M. V. Sharakhova, J. Kayondo, N. F. Lobo, F. Santolamazza, A. della Torre, F. Simard, F. H. Collins, and N. J. Besansky, "Breakpoint structure reveals the unique origin of an interspecific chromosomal inversion (2la) in the anopheles gambiae complex," *Proceedings of the National Academy of Sciences*, vol. 103, no. 16, pp. 6258–6262, 2006.

[20] N. F. Lobo, D. M. Sangaré, A. A. Regier, K. R. Reidenbach, D. A. Bretz, M. V. Sharakhova, S. J. Emrich, S. F. Traore, C. Costantini, N. J. Besansky, and F. H. Collins, "Breakpoint structure of the anopheles gambiae 2rb chromosomal inversion," *Malaria Journal*, vol. 9, Oct. 2010.

[21] D. M. Sangare, *Breakpoint Analysis of the Anopheles Gambiae S.S. Chromosome 2Rb, 2Rc, and 2Ru Inversions*. PhD thesis, University of Notre Dame.

[22] R. B. Corbett-Detig, I. Said, M. Calzetta, M. Genetti, J. McBroome, N. W. Maurer, V. Petrarca, A. della Torre, and N. J. Besansky, "Fine-mapping complex inversion breakpoints and investigating somatic pairing in the anopheles gambiae species complex using proximity-ligation sequencing," *Genetics*, vol. 213, no. 4, pp. 1495–1511, 2019.