Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Application Note

Genome Analysis

FGTpartitioner: Parsimonious delimitation of ancestry breakpoints in large genome-wide SNP datasets

Tyler K. Chafin[1*]

[1]Department of Biological Sciences, University of Arkansas, Fayetteville, AR 72701, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Partitioning large (e.g. chromosomal) alignments into ancestry blocks is a common step in phylogenomic analyses. However, current solutions require complicated analytical assumptions, or are difficult to implement due to excessive runtimes and unintuitive documentation. Additionally, most methods require haplotype phasing, which is often intractable for non-model studies. FGTpartitioner provides an efficient solution for non-model diploid genomes in which phasing is not possible, using minimal analytical assumptions and parallel computation.

Results: FGTpartitioner processes a full-chromosome alignment orders of magnitude faster than alternative solutions, while recovering identical results. It also provides flexible options for handling diploid data. Efficiency is further increased by using native parallelization, as well as intuitive chromosomal region targeting, which can be used to further distribute work for individual chromosomes or regions across multiple machines.

Availability: Complete code and documentation for FGTpartitioner are available as an open-source repository on GitHub: https://github.com/tkchafin/FGTpartitioner

Contact: tkchafin@uark.edu

1 Introduction

Inferring localized ancestries of different parts of diploid species' genomes has become a major goal in phylogenomic studies and is a critical step in reconstructing species trees (Springer and Gatesy, 2018). Analyzing genealogical variation in genomic datasets is necessary to understand processes such as horizontal transfer and hybridization, and has led to insights into adaptive introgression (Fontaine *et al.*, 2015), genome-wide selection (Sabeti *et al.*, 2002), disease susceptibility (Seldin *et al.*, 2011), and ancestral human demographics (Sankararaman *et al.*, 2014).

Multiple approaches have been proposed which allow for the delimitation of local ancestry blocks in the genome (i.e. establishing recombination breakpoints), but they generally fall into one of two categories: those which require dense or phased genotypic data (Liu *et al.*, 2013); and those with complex analytical assumptions which require the definition of informative prior probability distributions and are computationally intensive (Dutheil *et al.*, 2009). Both conditions are problematic for genome-scale studies of non-model diploid organisms, where large-scale resequencing and phased reference data are unavailable, and genomes are sequenced at low coverage.

I here describe a solution, FGTpartitioner, which is specifically designed for use with non-model genomic data without the need for high-quality phased reference data or dense population-scale sampling. FGTpartitioner delimits chromosome scale alignments using a fast approach which detects incompatible SNPs which violate the four-gametes assumption (Hudson and Kaplan, 1985), and rapidly resolves a minimal set of recombination events to produce a most parsimonious set of non-overlapping intervals which are both unambiguously defined and consistent regardless of processing order.

2 Algorithm and user-interface

For ease of application, inputs are required to follow the widely used VCF format (Danecek *et al.*, 2011). Users may provide parameter settings as arguments in the command-line interface which can restrict block delimitation to a certain chromosome (<-c> flag), with the option to additionally target a region via start (<-s>) and end (<-e>) coordinates. Parallel computation is also possible (<-t>) for particularly large alignments. After parsing user-inputs, the workflow of FGTpartitioner is as follows:

(1) For each SNP, perform four-gamete tests sequentially for rightward neighboring records, up to a maximal physical distance (if defined; <-d>) and stopping when a conflict (='interval') is found. Intervals are stored in a self-balancing tree. When using multiprocessing (<-t>), daughter processes are each provided an offset which guarantees a unique pairwise SNP comparison for each iteration

(2) Merge interval trees of daughter processes (if <-t 2 or greater>)

(3) Assign rank *k* per-interval, defined as the number of SNP records (indexed by position) spanned by each interval

(4) Order intervals by *k*; starting at min(*k*), resolve conflicts as follows: For each candidate recombination site (defined as the midpoint between each pair of spanned variants), compute the depth *d* of spanning intervals. The most parsimonious breakpoint is that which maximizes *d*

These algorithm choices have several implications: indexing SNPs by physical position guarantees that the same recombination sites will be chosen given any arbitrary ordering of SNPs; and defining breakpoints as physical centerpoints between nodes means that monomorphic sites will be evenly divided on either side of a recombination event. Because monomorphic sites by definition lack phylogenetic information, they cannot be unambiguously assigned to any particular ancestry block, thus my solution is to evenly divide them.

Heterozygous sites in diploid genomes are dealt with in multiple ways. By default, FGTpartitioner will randomly resolve haplotypes. The user can select an alternate resolution strategy (<-r>) which will either treat a SNP pair as failing if any resolution meets the four-gamete condition, or as passing if any possible resolution passes (i.e. the 'pessimistic' and 'optimistic' strategies of Wang *et al.*, 2010).

3 Benchmarking and comparison to other software

For testing the efficiency and scaling of this software, I ran tests using a 4-taxon alignment of *Canis lupus* chromosome 1 (~120 Mb). Raw reads were downloaded from the NCBI SRA (*C. lupus*: SRR7107787; *C. rufus*: SRR7107783; *C. latrans*: SRR1518489; *Vulpes vulpes*: SRR5328101-115), aligned with bowtie2 (Langmead and Salzberg, 2012) and variant-called using the HaplotypeCaller and best practices from GATK (McKenna *et al.*, 2010). Memory usage for processing a full alignment on 16 cores peaked at 18GB (with less memory required to run on less cores). Total runtime scales linearly with the maximum allowable physical distance between SNPs to check for four-gamete conditions, with <-d 250000> taking ~50 minutes and <-d 100000> taking 36 minutes to process an alignment comprising >2 million variants. Increasing number of cored offer diminishing returns, with 1, 2, 4, 8, and 16 cores taking 39, 20, 12 , 10, and 9 minutes, respectively (to process a 1 million base subset of the 4-taxon chromosome 1 alignment). Runtimes also scale linearly with dataset size (in total variants). Haplotype resolution strategy (random, pessimistic, or optimistic) was not found to impact runtimes.

For comparison, I attempted to delimit recombinatorial genes using an alternate method based on the four-gamete test, RminCutter.pl (https://github.com/RILAB/rmin_cut). Because RminCutter does not handle diploid data, I created pseudohaplotypes by randomly resolving heterozygous sites (pseudoHaploidize.py; https://github.com/tkchafin/scripts). I also ran the MDL approach (Ané, 2011), which uses a parsimony criterion to assign breakpoints separating phylogenetically homogenous loci, while penalizing for the number of breakpoints. Neither MDL nor RminCutter could complete an analysis on the full chromosome 1 dataset in a reasonable time span (capped at 10 days), so I first divided the alignment into blocks of ten-thousand parsimony-informative sites each (as these determine runtime, not total length). This produced 21 sub-alignments. MDL took 24 hours across 48 cores to complete a single 5 Mb segment, thus was not further explored. RminCutter.pl took an average of 22 hours per segment (single-threaded). Of note, RminCutter and FGTpartitioner yielded identical results when using comparable settings restricted to a pseudo-haploid dataset. FGTpartitioner thus offers an efficient and methodologically simple solution for ancestry delimitation in non-model diploid organisms with large genomes.

Acknowledgements

I would like to thank vonHoldt *et al.* (2016) and Kukekova *et al.* (2018) for making their raw datasets available via the NCBI SRA, and the Arkansas High Performance Computing Center and my Ph. D. advisors Drs. Michael and Marlis Douglas for access to computational resources which I used for developing and benchmarking this program.

Funding

This project was supported by computational resources provided by XSEDE Startup Allocation TG-BIO160058 and Research Allocation TG-BIO160065, and by the Arkansas High Performance Computing Center, which is funded by the Arkansas Economic Development Commission (Arkansas Settlement Proceeds Act of 2000).

Conflict of Interest: none declared.

References

Ané,C. (2011) Detecting phylogenetic breakpoints and discordance from genome-wide alignments for species tree reconstruction. *Genome Biol. Evol.*, **3**, 246–258.

Danecek,P. *et al.* (2011) The variant call format and VCFtools. *Bioinformatics*.

Dutheil,J.Y. *et al.* (2009) Ancestral population genomics: The coalescent hidden Markov model approach. *Genetics*, **183**, 259–274.

Fontaine,M.C. *et al.* (2015) Extensive introgression in a malaria vector species complex revealed by phylogenomics. *Science (80-.).*, **347**, 1258524.

Hudson,R.R. and Kaplan,N.L. (1985) Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, **111**, 147–164.

Kukekova,A. V *et al.* (2018) Red fox genome assembly identifies genomic regions associated with tame and aggressive behaviours. *Nat. Ecol. Evol.*, **2**.

Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie 2. *Nat. Methods*.

Liu,Y. *et al.* (2013) Softwares and methods for estimating genetic ancestry in human populations. *Hum. Genomics*, **7**, 1–7.

McKenna,A. *et al.* (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*

Sabeti,P.C. *et al.* (2002) Detecting recent positive selection in the human genome from haplotype structure. *Nature*, **419**.

Sankararaman,S. *et al.* (2014) The Genomic Landscape of Denisovan and Neanderthal Ancestry in Present-Day Humans. *Nature*, **507**, 354–357.

Seldin,M.F. *et al.* (2011) New approaches to disease mapping in admixed populations. *Nat. Rev. Genet.*, **12**, 523–528.

Springer,M.S. and Gatesy,J. (2018) Delimiting Coalescence Genes (C-Genes) in Phylogenomic Data Sets. 1–19.

vonHoldt,B.M. *et al.* (2016) Whole-genome sequence analysis shows that two endemic species of North American wolf are admixtures of the coyote and gray wolf. *Sci. Adv.*, **2**, e1501714–e1501714.

Wang,J. *et al.* (2010) Genome-wide compatible SNP intervals and their properties. *Proc. First ACM Int. Conf. Bioinforma. Comput. Biol.*, 43–52.