*NAR Genomics and Bioinformatics*, 2025, **7**, lqaf186
https://doi.org/10.1093/nargab/lqaf186
**Methods Article**

OXFORD

# HaploExplore, a software specifically designed for the detection of minor allele (MiA-) haploblocks

**Matilde Manetti [1,†], Samuel Hiet [1,†], Myriam Rahmouni [1], Jean-Louis Spadoni [1], Alice Dobiecki [1], Marco Lamanda [1], Maxime Tison [1], Taoufik Labib [1], Cristina Giuliani [2], Sigrid Le Clerc [1], Jean-François Deleuze [3], Jean-François Zagury [1,*]**

[1]Laboratoire Génomique, Bioinformatique, et Chimie Moléculaire, EA7528, Conservatoire National des Arts et Métiers, 2 rue Conté , Paris, 75003, France
[2]Laboratory of Molecular Anthropology, Department of Biological, Geological and Environmental Sciences, University of Bologna,Via Selmi 3, Bologna, 40126, Italy
[3]Laboratory for Genomics, Foundation Jean Dausset—CEPH, Paris, 75010, France

*To whom correspondence should be addressed. Email: zagury@cnam.fr
†The first two authors should be regarded as Joint First Authors.

## Abstract

Haplotype blocks in the genome are informative of evolutionary processes and they play a pivotal role in describing the genomic variability across human populations and susceptibility/resistance to diseases. Several software have been developed for haplotype block detection, but they do not distinguish between the impacts of major and minor single nucleotides polymorphism (SNP) alleles. In this study, we present a powerful haploblock detection software, specifically designed for identifying haploblocks associated with SNP minor allele haploblocks (MiA-haploblocks). These haploblocks are particularly important as they can significantly influence phenotypic traits, offering a novel approach for studying genetic associations and complex traits.

HaploExplore operates on VCF files containing phased data, exhibiting rapid processing times, and generating user-friendly outputs. Results converge when analyzing populations of 100 individuals or more. A comparative analysis of HaploExplore against other haploblock detection software revealed its superiority in terms of either simplicity, flexibility, or speed, with the unique capability to target minor alleles. HaploExplore will be very useful for evolutionary genomics and for GWAS analysis in human diseases, given that the effects of genetic associations may accumulate within a specific haploblock.

## Introduction

Haplotypes are combinations of variant alleles with substantial linkage disequilibrium (LD) within a genomic region of a population [1]. The set of haplotypes found in a genomic region can delineate haplotype blocks, LD blocks, or haploblocks, which provide valuable insights into evolutionary and recombinatory processes useful for applications ranging from population genetics to disease mapping [1, 2]. LD block regions vary in size, with strong LD persisting within blocks until interruption, often due to recombination hotspots or population genetic phenomena [3, 4]. Their patterns and position are shaped by recombination, mutation, selection, demography, and various other evolutionary forces [5].

Haploblocks are indispensable in genomic research and genome-wide association studies (GWAS) due to their ability to represent high-LD regions, allowing the identification of genomic areas linked to complex traits and disorders [6]. Moreover, population differences in LD blocks may be one of the reasons for replication issues in GWAS across different populations. Analyzing haploblocks improves understanding of gene regulation, epistatic interactions, and evolutionary patterns [7]. By grouping associated variants, haploblocks simplify LD structures, and may help reduce noise in GWAS analyses, and improve the statistical power to identify causal variants [8]. They offer a framework for understanding how genetic variations collectively influence phenotypic outcomes and biological processes and provide insights into population-specific genetic selection, admixture events, and ancestral recombination patterns.

In GWAS, haploblocks may also capture the cumulative effects of variants in LD, offering a broader perspective than single genetic markers. For example, in a recent study, our group has identified a 1.9 megabase haploblock of 376 single nucleotides polymorphisms (SNPs) associated with the allele HLA-B*57:01 [9]. By exploring the haploblock structure "containing" this HLA allele, we were able to identify SNPs having cumulative effects on the phenotype within the haploblock, explaining the major signal observed for HLA-B*57:01 for HIV-1 elite control [10]. By focusing on haploblocks, researchers can thus uncover interdependencies and synergistic effects often missed in single-variant approaches, revealing critical factors in disease resistance and susceptibility [10].

**Received:** June 4, 2025. **Revised:** November 4, 2025. **Accepted:** November 7, 2025

A critical issue not addressed by existing software is the need to focus on minor alleles. One can reasonably conceive that some low-frequency alleles have been selected in the course of evolution because they are associated to resistance to specific diseases, as has been described for the famous D32 mutant of the coreceptor CCR5 probably selected in Europeans for its better resistance to plague [11]. Conversely, some low-frequency alleles may be selected during evolution due to their benefits but also increased susceptibility to certain diseases, as has been shown for the $\beta^s$ allele (β-globin gene), which protects against malaria but acts as risk allele to sickle-cell disease [12]. Indeed, studies have shown that (i) minor alleles are more likely to be associated with disease susceptibility and resistance [13] and (ii) disease-associated alleles are more likely to be low-frequency derived alleles than neutral expectations [14]. Focusing on minor alleles thus becomes a useful strategy to look for genetic markers of disease resistance/susceptibility. In order to address the need for a robust analysis of minor alleles in haploblocks, we have developed the HaploExplore software, specifically designed to facilitate such investigations. It works with phased genotypes and can handle very large genomic regions and datasets, involving thousands of SNPs, with the capacity to build haploblocks composed of the SNPs minor allele (MiA-haploblocks). It relies on explicit parameters such as minor allele frequency (MAF), $r^2$, D′, and carrier percentage (CP), which makes it suitable for a variety of applications. Its usefulness for genomic analyses is completed by simple output visualizations.

## Materials and methods

### Test population

In this study, for sake of performance evaluation, we have used subgroups of a cohort genotyped with an Illumina chip, the DESIR (Data from an Epidemiological Study on Insulin Resistance Syndrome) cohort. It included 2576 men and 2636 women (aged 30–64 years at enrollment, 1994–1996) from the general French population [15], and a subcohort of ∼1500 subjects was genotyped using the Illumina Infinium II HumanHap300 BeadChips (Illumina, San Diego, CA). For the present study, we randomly selected 500 individuals and focused on chromosome 22. Data were imputed and phased using the Michigan Imputation Server with the HRC r1.1 2016 (GRCh37/hg19) reference panel, imputation followed a quality filter of $r^2 > 0.3$, and phasing was performed using Eagle v2.4 under the European (EUR) population model. After imputation, we obtained 125 956 SNPs on chromosome 22.

### Algorithm principle

The software processes a variant call format (VCF) file [16] (compressed or not) and extracts the useful information (e.g. SNP-ID, position), and applies a filter based on the SNP's MAF. To speed up the analysis and reduce computational complexity, the algorithm splits the genome into smaller regions, with the region size being user-defined. To build these subregions, the algorithm uses the sliding window approach, where overlapping subregions are created. The overlap size is equal to the maximum haploblock size parameter. The goal of the sliding window is to improve computational efficiency with preservation of linkage structure. Scanning an entire chromosome would require testing all chromosome-wide SNPs for membership in each candidate haploblock, which is computationally impractical with SNP counts exceeding 100 000. Empirically, a 10-Mb window with a 5-Mb overlap reliably captures MiA-haploblocks up to 5 Mb. Haploblocks larger than 5 Mb are relatively rare genome wide and can be handled subsequently by a focused, targeted analysis. Within each subregion, the SNPs are sorted in an ascending order of MAF, and the algorithm proceeds to scan the SNPs of the subregion according to this order. The algorithm starts from the first SNP of the ordered list, designated as the core SNP, serving as the anchor for the first haploblock. The coreSNP serves for identifying a group of SNPs whose minor alleles are genetically linked to its minor allele. The algorithm selects one by one the other SNPs (SNPtested) in the list and tries to add them in the haploblock according to the parameters. If the SNPtested is not too far from the coreSNP, based on the value of the maximum haploblock size parameter, then the algorithm checks whether the MAF of the SNPtested is greater than the MAF cut threshold of the coreSNP. If so, the algorithm computes $r^2$ and D′ between the coreSNP and the SNPtested, ensuring they meet the defined LD threshold. However, the classical LD parameters ($r^2$ and D′) show limitations in the specific identification of MiA-haploblocks. In particular, $r^2$ can be unreliable for SNPs with markedly different MAFs, while D′ does not adequately capture the co-occurrence of minor alleles (Supplementary Tables S2 and S3). To address this limitation, HaploExplore introduces a novel parameter: the CP. So, if the SNPtested reaches this step, the CP is computed in order to verify whether the minor allele of the SNPtested is frequently co-occurred with the coreSNP's minor allele, based on a predefined threshold. After the creation of haploblocks, the algorithm refines and corrects haploblocks by checking whether two consecutive SNPs within the block are too far apart, based on the position order. The default SNP gap threshold is 200 SNPs. Each genomic region is processed using one of three possible computational modes: the "Standard" mode, the "ListSNP" mode, and the "Exhaustive" mode.

Figure 1 summarizes the principal steps of the algorithm, the main parameters, the available computational modes, and the possible outputs (e.g. Supplementary Fig. S2).

### Computational modes

The "Standard mode" constructs haploblocks iteratively. Each time an SNP does not fit in existing haploblocks, a new block is created using this SNP as the coreSNP. With the chosen LD parameters, SNPs minor alleles are included in blocks if they meet the requested setting with regard to the coreSNP: LD between the SNP and the coreSNP, percentage carried by the coreSNP carriers, MAF percentage cut.

The "ListSNP mode" builds haploblocks based on a predefined list of SNPs, using the previously mentioned criteria. In this mode, haploblocks are initiated by SNPs present in the given list. These SNPs are considered as coreSNPs, and individual haploblocks are created for each SNP of the list. The user can set a first option in which SNPs from the list are used uniquely to generate the MiA-haploblocks, and no additional SNPs from the region will generate new haploblocks. Nevertheless, a second option allows the user to specify whether additional SNPs in the region can also be used to create new haploblocks (thus resembling the standard mode). With a second parameter, we can choose whether the SNPs in the list can also participate to other haploblocks (option: yes) or whether

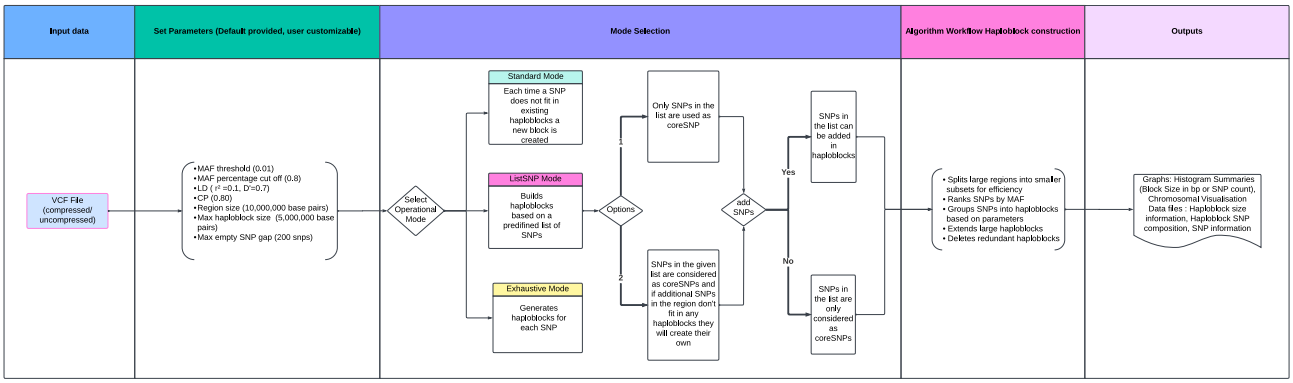


**Figure 1.** Workflow of HaploExplore, pipeline of construction of the haploblocks and outputs.

the SNPs in the list cannot be added in others haploblocks and are thus only used as coreSNPs (option: no). Thus, the ListSNP mode is especially useful for focused analyses, such as studying specific loci of interest (e.g. HLA regions). It is possible to specify criteria such as LD thresholds, MAF, and CP cutoffs for refining haploblocks.

The "Exhaustive mode" aims to reconstruct haploblocks by iteratively scanning through all SNPs in a region. This iterative mode starts by assigning to each SNP its own haploblock. It employs LD and CP criteria to refine haploblocks while maintaining meaningful boundaries. This mode is particularly useful if the aim is to reconstruct haploblocks in regions with complex LD patterns or high genomic diversity.

## Main parameters

In the following paragraph coreSNP and SNPtested are referred as SNP1 and SNP2, respectively. Many parameters can be considered to identify haplotype blocks within a genomic region, among them, critical ones are : (i) MAF, SNPs are selected ensuring that the variants are common enough to be informative for block definition; (ii) LD thresholds, such as $r^2$ and D′, to make sure SNPs are sufficiently correlated with the coreSNP of the haploblock [17, 18]; (iii) CP, this parameter is used to include an SNP2 minor allele in a haploblock when a sufficient proportion of individuals carrying a given SNP1 minor allele (coreSNP of the haploblock) also carry SNP2 minor allele. This ensures that SNPs are grouped with SNP1 when they are frequently observed together in individuals, allowing the haploblock to better represent their potential combined influence on genetic variation. This parameter ensures that the constructed haploblocks fulfill the criteria of MiA-haploblocks by grouping SNPs based on their co-occurrence of minor alleles.

The CP formula can be expressed as follows:

$$\mathrm{CP}\,(\mathrm{SNP1},\,\mathrm{SNP2}) = \frac{N_{\mathrm{SNP1},\,\mathrm{SNP2}}}{N_{\mathrm{SNP1}}} \times 100,$$

where

- $N_{\mathrm{SNP1},\,\mathrm{SNP2}}$ is the number of individuals carrying the minor allele of SNP2 who also carry the minor allele of SNP1 (coreSNP);
- $N_{\mathrm{SNP1}}$ is the total number of individuals carrying the minor allele of SNP1

Since we are referring to phased data, the CP is computed separately for each chromosome copy (i.e. each chromatid), allowing for more accurate haploblocks construction:

$$\mathrm{CP}_1\,(\mathrm{SNP1},\,\mathrm{SNP2}) = \frac{N_{\mathrm{SNP1},\,\mathrm{SNP2}\,(1)}}{N_{\mathrm{SNP1}\,(1)}} \times 100,$$
$$\mathrm{CP}_2\,(\mathrm{SNP1},\,\mathrm{SNP2}) = \frac{N_{\mathrm{SNP1},\,\mathrm{SNP2}\,(2)}}{N_{\mathrm{SNP1}\,(2)}} \times 100,$$

where

- $N_{\mathrm{SNP1},\,\mathrm{SNP2}\,(1)}$ and $N_{\mathrm{SNP1},\,\mathrm{SNP2}\,(2)}$ are the number of individuals carrying the minor allele of SNP2 who also carry the minor allele of SNP1 on chromatid 1 and 2, respectively.
- $N_{\mathrm{SNP1}(1)}$ and $N_{\mathrm{SNP1}\,(2)}$ are number of individuals carrying the minor allele of SNP1 on chromatid 1 and 2, respectively.

The overall exact CP across both chromatids is:

$$\mathrm{CP}_{\mathrm{exact}}\,(\mathrm{SNP1},\ \mathrm{SNP2}) = \mathrm{CP}_1\,(\mathrm{SNP1},\ \mathrm{SNP2}) + \mathrm{CP}_2\,(\mathrm{SNP1},\ \mathrm{SNP2}).$$

This ensures that only alleles co-occurring on the same haplotype are counted together. Consequently, CPexact is not simply equal to 2 × CP1, except in the special case where minor alleles are distributed symmetrically across chromatids. By considering CP values separately for each chromatid (CP1 and CP2), our approach captures haplotype asymmetries that would otherwise be overlooked in unphased data, leading to a more accurate detection of MiA-haploblocks.

(iv) The MAF cut threshold is defined as a relative threshold based on the MAF of the coreSNP to determine which SNPs can be included in a haploblock. To be clear:

$$\mathrm{MAF}_{\mathrm{cut}} = \alpha_{\mathrm{cut}\,\%} \times \mathrm{MAF}_{\mathrm{coreSNP}}.$$

where:

- $\mathrm{MAF}_{\mathrm{cut}}$ is the minimum MAF required for an SNPtested to be included in the haploblocks;
- $\alpha_{\mathrm{cut}\,\%}$ is the MAF percentage cut, the actual tunable parameter in the software;
- $\mathrm{MAF}_{\mathrm{coreSNP}}$ is the MAF of the coreSNP.

Default values for these parameters are provided but can be easily modified by the user. For additional details, see "Section 1.1: Default Settings" in Supplementary data (Supplementary Table S1 and Supplementary Fig. S1).

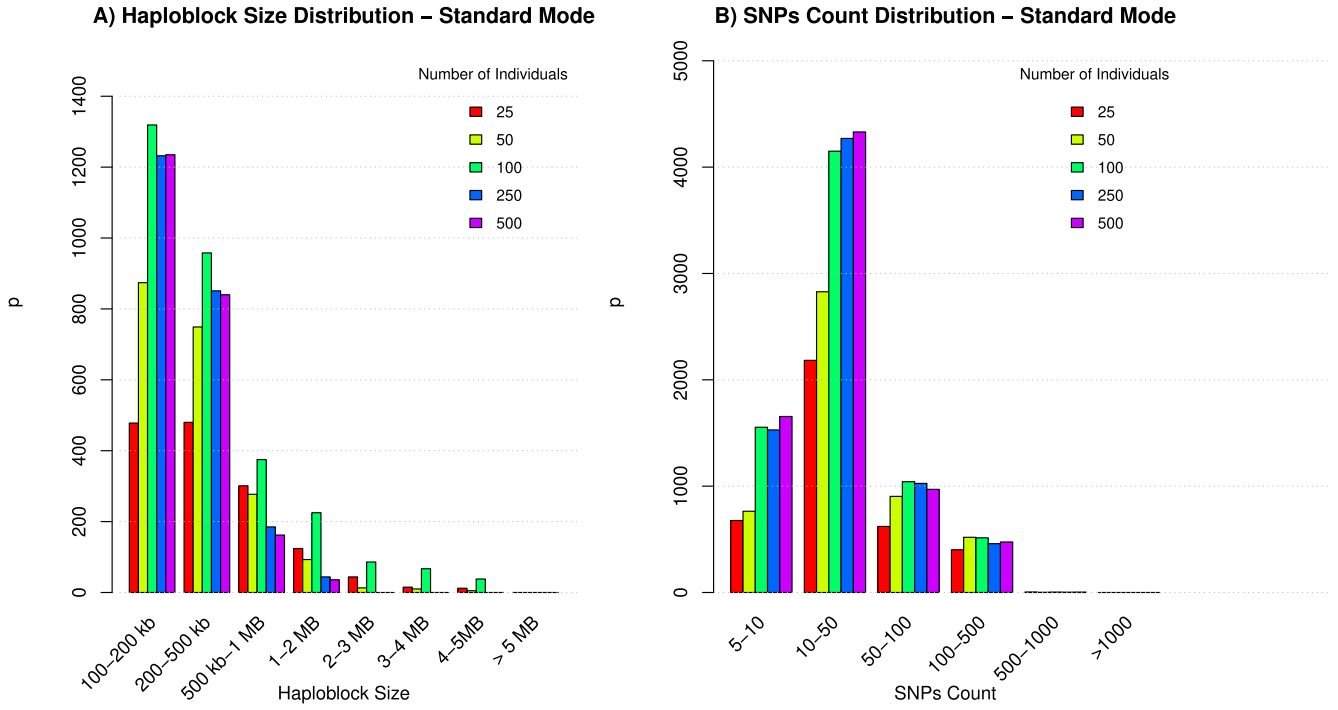


**Figure 2.** Haploblock and SNP count distributions for standard mode. (**A**) Distribution of haploblock sizes across various sample sizes from the DESIR cohort. The *x* axis displays haploblock size categories, ranging from 100 kb to over 5 MB, while the *y* axis represents the number of haploblocks observed in each size category. Bar plots are used to show the distribution for each sample size (25, 50, 100, 250, 500 individuals), with distinct colors representing different sample sizes. The reduced plot excludes the smallest haploblock size category (0–100 kb) to enhance the visualization of larger haploblocks. (**B**) Distribution of SNP counts per haploblock across various sample sizes in the analysis. The *x* axis shows SNP count categories (ranging from 5 to >1000 SNPs per haploblock), while the *y* axis represents the number of haploblocks observed. Bar plots are used to display the results for all sample sizes (25, 50, 100, 250, 500 individuals), with each sample size represented by a unique color. The reduced plot excludes the categories 1, 2, 3, and 4 SNPs to provide a clearer view of the distribution for haploblocks with higher SNP counts. Both panels are based on data from chr 22 (125 956 SNPs in 35 MB) using the standard mode of HaploExplore. Parameters include: LD thresholds ($r^2 \geq 0.1$, $D' \geq 0.7$), MAF cut off ≥0.8, %carrier threshold (CP) ≥80%, region size of 10 000 000 base pairs with a region overlap of 5 000 000 base pairs, and a maximum empty gap of 200 SNPs.

## Results

### Impact of the population size

To evaluate the impact of sample size on haploblock detection and convergence, we used phased and imputed SNPs in the 35-Mb-long chromosome 22 involving 125 956 SNPs data from the DESIR cohort of French individuals. Analyses were conducted on subsets of 25, 50, 100, 250, and 500 individuals, and two key outputs were assessed: the distribution of haploblock sizes in bp and the number of SNPs per haploblock (Fig. 2A and B).

Results demonstrate that smaller sample sizes (between 25 and 100 individuals) produce more variable outputs, with wider fluctuations in both haploblock size and SNP counts (Fig. 2A and B). This variability reflects a lack of convergence, likely driven by insufficient sample representation of LD patterns. The results indicate a clear trend toward convergence, with diminishing changes beyond 100–250 individuals (Fig. 2A and B). Additional details, including extended tables and figures supporting these findings, are provided in the "Section 2.1: Convergency" of Supplementary data (Supplementary Tables S4–S6). A simplified visualization of the results is presented in Fig. 2. Very small haploblocks (<100 kb or containing ≤4 SNPs) were excluded because their high frequency would have dominated the plots and obscured the distribution of larger, more biologically informative haploblocks. This adjustment improves clarity and highlights patterns in the larger haploblocks, which are more likely to capture meaningful linkage structures. For completeness, the full results, including all haploblock size and SNP count categories, are provided in the Supplementary Materials "Section 2.2: Impact of population size" of Supplementary data (Supplementary Figs S3–S5). For these analyses, we have employed the standard mode for haploblock detection, with the default parameters. While the computational speed analysis will be addressed later, these first findings highlight the importance of an adequate sample size to achieve convergence and reliable detection of haploblock boundaries.

### Benchmark validation

Additionally, the Supplementary data present a comparison of chromosome 22 haploblocks between two French and two African–American cohorts (Supplementary Fig. S6), and a comparison of haploblocks on chromosomes 1, 6, and 22 within the DESIR cohort (Supplementary Fig. S7). These analyses confirm the consistency of the MiA–haploblocks within a population and across different chromosomes.

We also verified that HaploExplore retrieved the manually computed HLA haploblocks described in our previous work dealing with the role of MHC region SNPs in HIV–1 elite control [10], and they were indeed fully reproduced (Supplementary Table S7).

### Speed

The computational efficiency of a software is always an important factor, given the challenges posed by large sample sizes

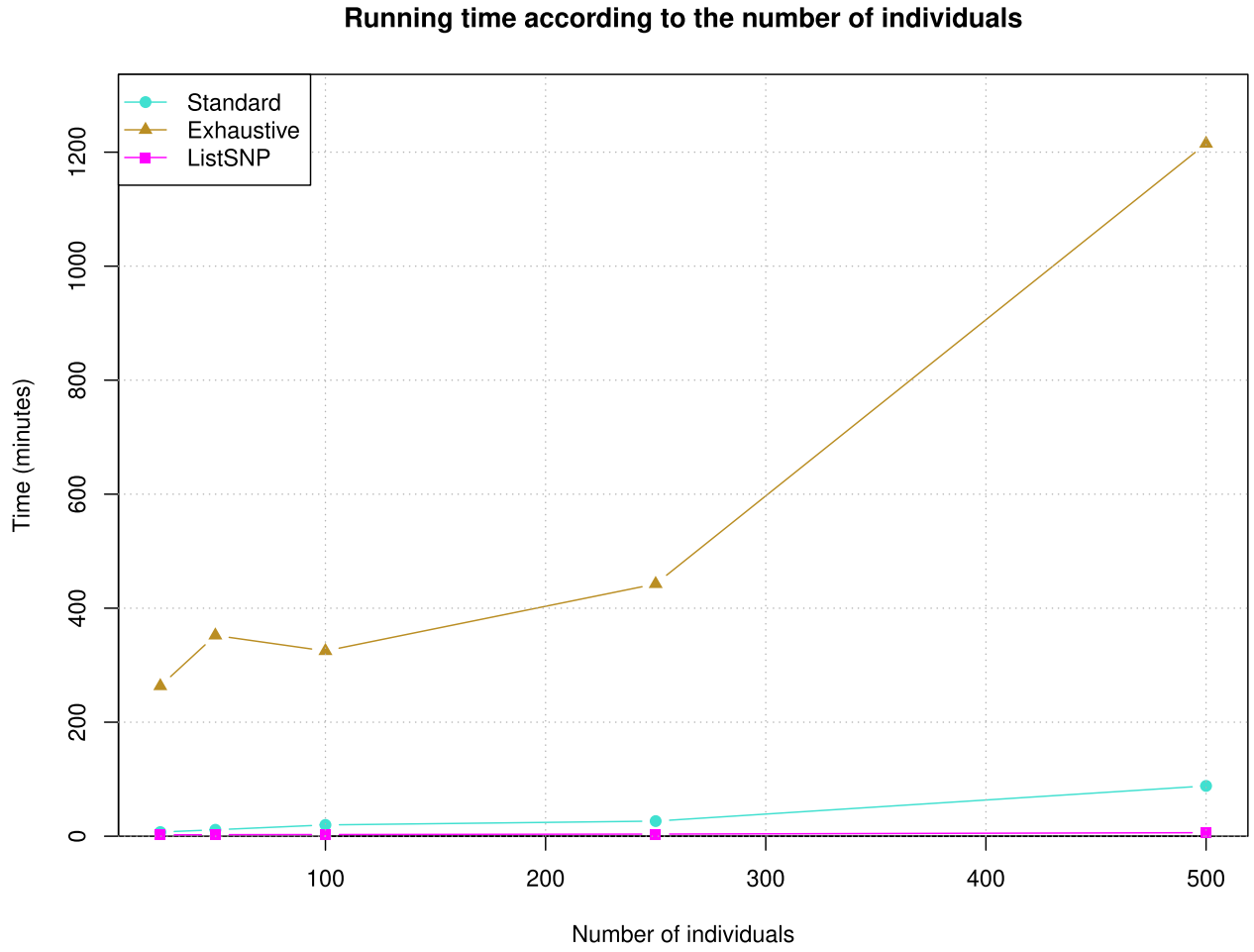


**Figure 3.** Running time of HaploExplore across different population sizes (25, 50, 100, 250, and 500 individuals from DESIR cohort) for the Standard mode with LD, Exhaustive mode, and ListSNP mode on chr 22 (size of 35 Mb) 125 956 SNPs.

and extensive datasets. We evaluated the running time of the three processing modes (ListSNP, Standard, and Exhaustive) across varying sample sizes (25, 50, 100, 250, and 500 individuals) (Fig. 3 and Supplementary Table S8).

The running times were quite reasonable since they were in terms of minutes to hours according to the mode. For instance, for both phased and imputed data, the ListSNP mode with at list of 500 SNPs exhibited running times of a few minutes across all sample sizes (chromosome 22: 125 956 SNPs in 35 Mb) with only 2 min for 25 individuals and 6 min for 500 individuals (Supplementary Table S10). By contrast, the exhaustive mode, which computes haploblocks for all SNPs in a region without relying on predefined lists, was significantly slower due to the increased computational demand, always considering the same chromosome 22. This mode processes a much larger number of SNPs, consequently, the exhaustive mode had running times ranging from 263.41 min for 25 individuals to 1215 min (around 20 h) for 500 individuals over the chromosome 22 (Supplementary Table S9).

The Standard mode was slower than the ListSNP mode but considerably faster than the Exhaustive mode. These results illustrate the trade-offs between computational speed and analysis scope. ListSNP is ideal for analyses targeting a specific set of SNPs, offering high efficiency and speed. However, the Exhaustive mode remains indispensable for comprehensive studies that require analyzing all SNPs in a genomic region. The choice of the mode depends on the scope of the study of interest and available computational resources.

It is important to note that runtime also scales with the number of SNPs per chromosome. For example, analyses on larger chromosomes (e.g. chromosome 1) naturally take longer than those on smaller ones (e.g. chromosome 22), even under the same sample size and mode (Section 2.6: Speed, Supplementary data). Thus, there is a proportional relationship between SNP count and runtime.

### Comparison with other haploblock detection software

Several computational tools exist for haploblock detection, each employing distinct methodologies and having differences in parameter flexibility. In Table 1, we compared their algorithm, parameters, and other features with those of HaploExplore. Additional details are shown in "Section 2.7: Comparison with other haploblocks detection software" of Supplementary data (Supplementary Tables S11–S13).

## Discussion

A new tool for haploblock detection has been developed, specifically designed to target minor alleles. It provides flexibility and precision by implementing the definition of haploblocks based on multiple biological and statistical parameters, including MAF, LD measures ($r^2$ and D′), and CP. Its direct processing of standard VCF files ensures compatibility with widely used genomic data. HaploExplore manages effectively large genomic datasets by partitioning data into smaller

**Table 1.** Comparison of haploblock detection tools: summary of key features across different haploblock detection tools, including input data formats, haploblock construction methods, parameter flexibility, output types, and support for MiA-haploblocks and programming language

| Tool | Input data | Haploblock construction | Flexibility (parameters) | Output data | MiA-haploblocks | Programming language |
|---|---|---|---|---|---|---|
| HaploExplore | VCF (compressed or uncompressed) | Uses $r^2$ and D′, CP with tunable cutoffs | High (MAF, LD thresholds ($r^2$, D′), CP, max haploblock size, region split, MAF percentage cut, maximum SNP gap) | Haploblock composition, data (txt files), statistics, graphical output with haploblocks boundaries | Yes | Python |
| PLINK | PED/MAP, BED/BIM/FAM | Uses LD approach of Gabriel's method, confidence interval of D′ | Medium [max block size, min MAF, LD confidence interval thresholds (low/high), recombination confidence interval, informative pair fraction] | Haploblock structure, detailed blocks file | No | C |
| Big-LD | Numeric genotype matrix | Graph-based LD segmentation: detects cliques based on \|r\| and partitions SNPs using an interval graph approach | Low [four parameters: LD \|r\| cutoff (CLQcut), max SNP gap (clstgap), sub-region boundary constraints (leng), max SNPs per segment (subSegmSize)] | Haploblocks with startSNP endSNP and position in bp | No | HTML/R |
| HaploBlocker | VCF; PED/MAP; genotype matrix object | Iterative clustering, merging, and filtering process using local haplotype similarities | Extremely high: ∼70 tunable parameters (window size, merging error, haplotype filtering, node/edge constraints, block extension, and more) | Haploblock composition and allow also visualization of the haploblocks | No | R/C++ |

genomic regions thereby allowing computational efficiency. It has the capacity to analyze an entire chromosome (>125 000 SNPs) in <10 mn. The software is not limited by population size; its running time scales linearly with the number of individuals analyzed (Supplementary Fig. S8). In our examples, we used up to 500 individuals as results were observed to stabilize starting with sample sizes of ∼100–250 individuals.

Its modes of operation (Standard mode, ListSNP mode, and Exhaustive mode) provide diverse approaches to build and refine haploblocks based on user-defined thresholds and specific research objectives. The Exhaustive and ListSNP modes are particularly adapted for exploratory analyses and targeted research on small genomic regions. The Exhaustive mode is especially beneficial when researchers aim to analyze each single nucleotide polymorphism individually within a biologically relevant region, while the ListSNP mode is tailored for scenario where a predefined list of SNPs is prioritized for MiA-haploblock analyses of specific genomic loci. In contrast, the Standard mode is designed for the analysis of entire genomes and the identification of haploblocks across extensive regions with varying SNP densities, making it the preferred choice for comprehensive genome exploration due to its scalability with increasing sample sizes.

All analyses in this study were performed on cohorts of unrelated individuals. However, HaploExplore is equally applicable to related individuals or to cohorts defined by specific phenotypes (e.g. disease status), depending on the study objectives.

The findings from our analysis of HaploExplore highlight the critical impact of sample size on haploblock detection and the robustness of different detection modes. Our results indicate that increasing the population size to 100 individuals or higher significantly enhances convergence, leading to stable and reproducible haploblock distributions. The computational performance analysis highlights the scalability of HaploExplore. While the Exhaustive mode remains the most time- and memory-consuming, the Standard and ListSNP modes offer more efficient alternatives, with execution times remaining reasonable even for large genomic datasets.

Different haploblock detection tools employ distinct methodologies, affecting their ability to capture MiA-haploblocks. PLINK and Big-LD define haploblocks based on traditional LD measures (such as D′, $r^2$), making them fitted for common haplotype structures but limiting their ability to detect MiA-haploblocks. HaploBlocker, using sequence haplotype similarity-based clustering, prioritizes long, stable haplotypes but does not focus on minor allele inheritance patterns. HaploExplore uniquely integrates LD thresholds with a CP criterion, allowing the identification of MiA-haploblocks, which are crucial for studying genetic associations and complex traits. These haploblocks capture co-inherited minor alleles that may influence disease susceptibility and trait variability; patterns often missed by LD-based methods. Moreover, the software provides an interactive environment, allowing users to visualize and adjust parameters before each analysis, improving usability and supporting exploratory analysis.

While HaploExplore has demonstrated robust performance and speed in haploblock detection in its current form, the modular structure of the programme makes it adaptable to allow for improvements answering to specific situations.

To conclude, HaploExplore is a sophisticated and flexible software solution for haploblock detection and analysis, with

a focus on minor alleles, offering a wide range of strengths that make it well suited for genomic research.

## Web resources—software availability

The software is available on GitHub at https://github.com/HaploExplore/HaploExplore and on Figshare at https://doi.org/10.6084/m9.figshare.30518498.

Additionally, we have developed an interactive web application using Streamlit, allowing users to run the software with a user-friendly interface. Default parameters are preset but can be adjusted directly from the application. A detailed description of the parameters, input, and output files is provided in the ReadMe.txt file.

## Acknowledgements

The Laboratory GBCM and the Foundation Jean Dausset-CEPH are grateful to the program Mécénat-Santé of Mutuelles AXA for funding this research.

*Author contributions*: Conceptualization: J.F.Z.; Data curation: T.L., S.L.C.; Formal analysis: M.M., S.H., M.R., A.D., J.F.Z.; Funding acquisition: J.F.D., J.F.Z.; Investigation: M.M., S.H., M.R.; Methodology: M.M., S.H., M.L., M.T., S.L.C., J.F.Z.; Project administration: J.F.D., J.F.Z.; Resources: T.L., S.L.C., J.F.D., J.F.Z.; Software: M.M., S.H., A.D., J.F.Z.; Supervision: C.G., J.F.Z.; Validation: M.M., S.H., M.R., J.L.S., S.L.C.; Visualization: M.M., S.H., A.D.; Writing—original draft: M.M., S.H., J.F.Z.; Writing—review & editing: All authors.

## Supplementary data

Supplementary data is available at NAR Genomics & Bioinformatics online.

## Conflict of interest

None declared.

## Funding

Matilde Manetti is recipient of a PhD fellowship from the Ministère de l'Enseignement Supérieur et de la Recherche. Samuel Hiet is recipient of a PhD fellowship from program Mécénat-Santé of Mutuelles AXA.

## Data availability

Genomic data used in this study cannot be transferred to the public but can be obtained under specific agreements upon contacting the corresponding author.

## References

1. Gabriel SB, Schaffner SF, Nguyen H *et al.* The structure of haplotype blocks in the Human genome. *Science* 2002;296:2225–29. https://doi.org/10.1126/science.1069424
2. Sabeti PC, Reich DE, Higgins JM *et al.* Detecting recent positive selection in the Human genome from haplotype structure. *Nature* 2002;419:832–37. https://doi.org/10.1038/nature01140
3. Jeffreys AJ, Kauppi L, Neumann R. Intensely punctate meiotic recombination in the class II region of the major histocompatibility complex. *Nat Genet* 2001;29:217–22. https://doi.org/10.1038/ng1001-217
4. Stephens M, Scheet P. Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation. *Am J Hum Genet* 2005;76:449–62. https://doi.org/10.1086/428594
5. McVean GAT, Myers SR, Hunt S *et al.* The fine-scale structure of recombination rate variation in the human genome. *Science* 2004;304:581–84. https://doi.org/10.1126/science.1092500
6. Cuyabano BCd, Su G, Lund MS. Selection of haplotype variables from a high-density marker map for genomic prediction. *Genet Sel Evol* 2015;47:61. https://doi.org/10.1186/s12711-015-0143-3
7. Traherne JA. Human MHC architecture and evolution: implications for disease association studies. *Int J Immunogenet* 2008;35:179–92. https://doi.org/10.1111/j.1744-313X.2008.00765.x
8. Karkar S, Dandine-Roulland C, Mangin J-F *et al.* Genome-wide haplotype association study in imaging genetics using whole-brain sulcal openings of 16,304 UK biobank subjects. *Eur J Hum Genet* 2021;29:1424–37. https://doi.org/10.1038/s41431-021-00827-8
9. Rahmouni M, De Marco L, Spadoni J-L *et al.* The HLA-B*57:01 Allele corresponds to a very large MHC haploblock likely explaining its massive effect for HIV-1 elite control. *Front Immunol* 2023;14:1305856. https://doi.org/10.3389/fimmu.2023.1305856
10. Rahmouni M, Le Clerc S, Spadoni J-L *et al.* Deep analysis of the major histocompatibility complex genetic associations using covariate analysis and haploblocks unravels new mechanisms for the molecular etiology of elite control in AIDS. *BMC Immunol* 2025;26:1. https://doi.org/10.1186/s12865-024-00680-6
11. Duncan SR. Reappraisal of the historical selective pressures for the CCR5-32 mutation. *J Med Genet* 2005;42:205–8. https://doi.org/10.1136/jmg.2004.025346
12. Laval G, Peyrégne S, Zidane N *et al.* Recent adaptive acquisition by African rainforest hunter-gatherers of the late pleistocene sickle-cell mutation suggests past differences in malaria exposure. *Am J Hum Genet* 2019;104:553–61. https://doi.org/10.1016/j.ajhg.2019.02.007
13. Kido T, Sikora-Wohlfeld W, Kawashima M *et al.* Are minor alleles more likely to be risk alleles?. *BMC Med Genet* 2018;11:3. https://doi.org/10.1186/s12920-018-0322-5
14. Lachance J. Disease-associated alleles in genome-wide association studies are enriched for derived low frequency alleles relative to HapMap and neutral expectations. *BMC Med Genet* 2010;3:57. https://doi.org/10.1186/1755-8794-3-57
15. Limou S, Coulonges C, Herbeck JT *et al.* Multiple-cohort genetic association study reveals CXCR6 as a new chemokine receptor involved in long-term nonprogression to AIDS. *J Infect Dis* 2010;202:908–15. https://doi.org/10.1086/655782
16. Danecek P, Auton A, Abecasis G *et al.* The variant call format and VCFtools. *Bioinformatics* 2011;27:2156–58. https://doi.org/10.1093/bioinformatics/btr330
17. Hill WG, Robertson A. Linkage disequilibrium in finite populations. *Theor Appl Genet* 1968;38:226–31. https://doi.org/10.1007/BF01245622
18. Lewontin RC. The interaction of selection and linkage. I. General considerations; heterotic models. *Genetics* 1964;49:49–67. https://doi.org/10.1093/genetics/49.1.49

**Received:** June 4, 2025. **Revised:** November 4, 2025. **Accepted:** November 7, 2025