Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: XXX XXX XXXX
XXXXXXXXXXXX

Subject Section XXXX

HAPLOSCOPER: A Comprehensive Pedigree Drawing and Haplotype Visualisation Web Application

Mehmet Tekman [1], Alan Medlar [2], Monika Mozere [1], Robert Kleta [1]*, and Horia Stanescu [1]

[1] Division of Medicine, University College London, London, NW3 2PF, UK and
[2] Institute of Biotechnology, University of Helsinki, Helsinki, 00014, Finland.

* To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Haplotype reconstruction is an important tool for understanding the aetiology of human disease. Haplotyping infers the most likely phase of observed genotypes conditional on constraints imposed by the genotypes of other pedigree members. The results of haplotype reconstruction, when visualised appropriately, show which alleles are identical by descent despite the presence of untyped individuals. When used in concert with linkage analysis, haplotyping can help delineate a locus of interest and provide a succinct explanation for the transmission of the trait locus. Unfortunately, the design choices made by existing haplotype visualisation programs do not scale to large numbers of markers. Indeed, following haplotypes from generation to generation requires excessive scrolling back and forth. In addition, the most widely-used program for haplotype visualisation produces inconsistent recombination artefacts for the X chromosome.

Results: To resolve these issues, we developed HAPLOSCOPER, a novel web application for haplotype visualisation and pedigree drawing. HAPLOSCOPER takes advantage of HTML5 to be fast, portable and avoid the need for local installation. It can accurately visualise autosomal and X-linked haplotypes from both outbred and consanguineous pedigrees. Haplotypes are coloured based on identity by descent using a novel A* search algorithm and we provide a flexible viewing mode to aid visual inspection. HAPLOSCOPER can currently process haplotype reconstruction output from Allegro, Genehunter, Merlin and Simwalk.

Availability: HAPLOSCOPER is licensed under GPLv3 and is hosted and maintained via Bitbucket.

Web Application and Source Code: `https://www.bitbucket.io/mtekman/haplo_html5`

Supplementary information: Supplementary data is available from *Bioinformatics* online.

Contact: r.kleta@ucl.ac.uk

1 Introduction

Linkage analysis, together with haplotype reconstruction, is used to identify putative locations of disease traits. Linkage analysis tests whether a given gene region co-segregates with the trait locus, whereas haplotype reconstruction infers the phase information that is lost during genotyping, i.e. the parental origin of each allele. In doing so, regions of interest can be found using linkage analysis and those regions delineated with inferred recombinations from haplotype reconstruction. Once a region has been identified, candidate genes can be selected for sequencing based on information from sequence databases (tissue-specific expression, homology, etc) or, if no candidate presents itself, all genes from the identified region can be screened for mutations using, for example, exome sequencing (Bockenhauer *et al.*, 2012).

Many parametric linkage analysis programs also perform haplotype reconstruction based on maximum likelihood. However, to integrate these analyses together requires advanced visualisation methods to intuitively

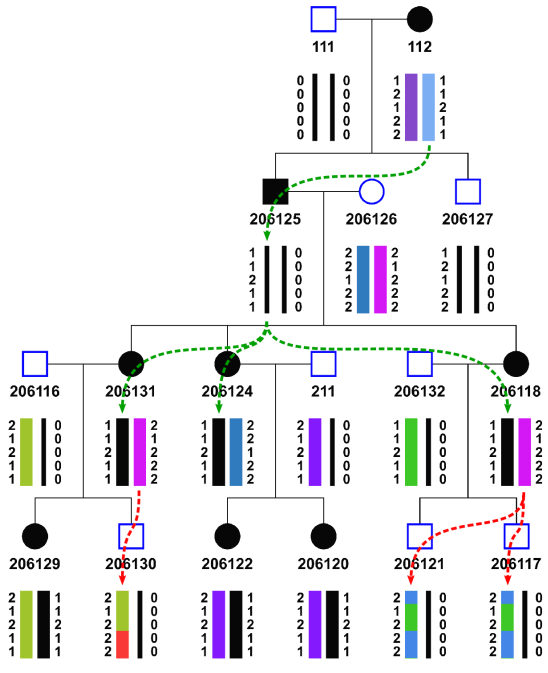

Fig. 1. HaploPainter visualisation of a five marker X-linked analysis. Colours indicate identity by descent. Arrows are overlayed to show the true flow of genetic data based on genotypes, with green showing inconsistent colouring between successive meioses and red showing erroneous inheritance.

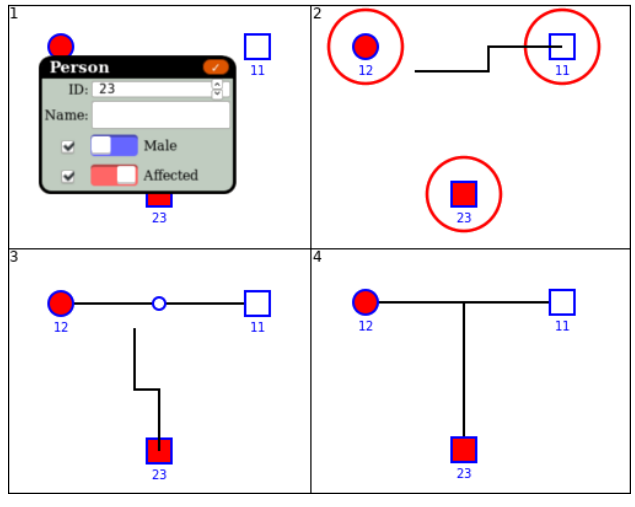

Fig. 2. Pedigree drawing view show the four stages of creating a pedigree: (1) Adding individuals and modifying their properties, (2) Joining mates with a mateline to anchor points made visible with red circles, (3) Joining offspring to their parents through a childline to anchor points made visible with white circles, (4) Completing a trio.

display haplotypes together with the pedigree structure and to colour haplotypes based on identity by descent (IBD).

There are many programs available for visualization, with HaploPainter being the most highly cited (Thiele and Nürnberg, 2005). However, our experience with HaploPainter has shown that viewing haplotypes in-line with the pedigree does not scale to large numbers of markers. Indeed, to compare haplotypes between generations requires excessive scrolling and the user has to re-identify the same region of interest over and over again in each successive generation. In addition, HaploPainter does not always correctly display which alleles are IBD, creating inconsistent recombination artefacts for the X chromosome. As shown in Figure 1, the last generation of individuals (particularly 206130, 206121 and 206117) appear to have undergone multiple recombinations within a relatively short genetic distance (< 1 cM). HaploPainter fails to properly account for the fact males have only a single X chromosome and, therefore, can appear to inherit from their father or even from an undisplayed paternal allele (see blue allele in individuals 206121 and 206117 in Figure 1).

To resolve these issues we present HAPLOSCOPER, a novel web application for haplotype visualization and pedigree drawing. HAPLOSCOPER is designed specifically for navigating high numbers of markers, providing a more intuitive viewing mode compared to other programs. Secondly, we present a novel A* search-based method that ensures IBD information is correctly displayed for all chromosomes, including the X chromosome. Finally, HAPLOSCOPER is web-based and therefore runs in any HTML5-compatible web browser and does not require local installation. Despite being web-based, it is fast and the user experience is similar to a native application, utilizing menus and a drag and drop interface.

2 Approach

HAPLOSCOPER is a comprehensive web application for haplotype visualisation and pedigree drawing. Here we will enumerate and expand upon the core features.

2.1 Pedigree Drawing

Pedigrees are drawn with a simple drag and drop interface, and are compliant with the Pedigree Standardization Work Group (PSWG) specification (Bennett *et al.*, 1995, 2008). The standard is already familiar to clinicians and allows individuals in the pedigree to be annotated with patient metadata.

Individuals are added to the pedigree either through context-dependent sidebars or user-customizable keyboard shortcuts, and the properties of that individual (sex, affection status, etc) can then be edited from a dialog box. Relationships between individuals are added by drawing *matelines* and *childlines*. Matelines indicate marriages and childlines connect children to their parents' mateline. Lines snap to context-dependent anchor points that become visible when adding relationships (Figure 2). Both members of each mateline are vertically aligned with one another and move together as a single unit. Siblings bound to the same mateline are similarly aligned automatically.

Projects can contain multiple families, and complex consanguineous relationships are automatically detected and represented with double-lines. Pedigrees can be loaded from and saved to local browser storage. Pedigrees can be imported and exported in standard LINKAGE (pre-makeped) format.

2.2 Haplotype Visualisation

While HaploPainter visualises haplotypes in-line with the pedigree, in HAPLOSCOPER haplotypes are displayed in a separate viewing mode. Selected individuals are aligned horizontally and grouped by family. Haplotypes are displayed underneath each individual, allowing for side-by-side comparison, irrespective of generation. The relatedness of individuals can be optionally displayed, with relationship lines stating their degree of separation via switcheable modes of pedigree alignment (see Figure 3).

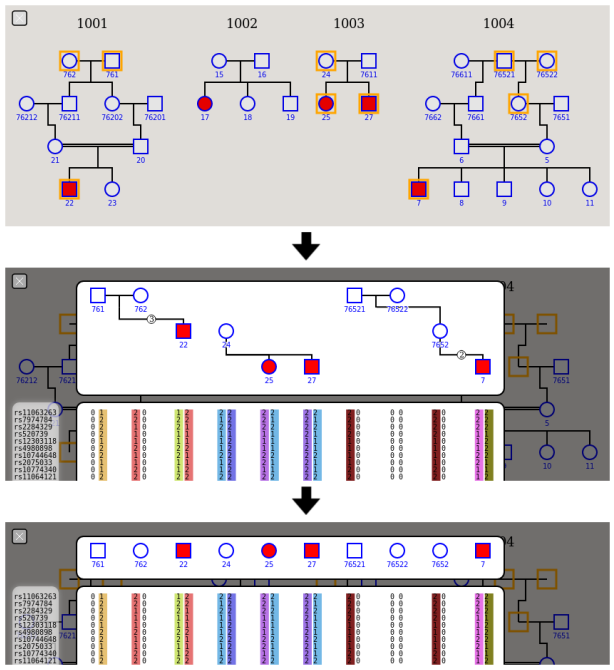

Fig. 3. (Top) Selection view enabling the subselection of individuals across multiple pedigrees. (Middle) Comparison view displaying the haplotypes of selected individuals, connected by lines indicating their degrees of separation from one another. (Bottom) Comparison view representing the same genotypes but with individuals vertically aligned.

Haplotypes are displayed within a viewport that defines the locus of interest across the genotypes of all selected individuals, the contents of which are outlined on the chromosome overview. The overview consists of a red vertical bar representing the entire length of the chromosome, and it is overlayed with a region indicator which has a height and position that maps to the size and location of the viewport on the chromosome.

The viewport can be manipulated by a variety of different interactions as shown in Figure 4: dragging the region indicator in the overview, scrolling the mousewheel over the genotypes, or using keyboard shortcuts for fast (PageUp/PageDn) and slow (Up/Down arrows) movement. Precise modes of adjustment are also provided; dragging and releasing haplotypes at a desired vertical position, shifting to the next/previous point of recombination, or specifying a pair of markers from a dropdown list.

The chromosome overview also allows for the size of the viewport to be adjusted through dragging the top and bottom handles on the region indicator as shown in Figure 4 (Bottom). By default, the size of the viewport is locked to the size of the window. However, if the viewport is made bigger than the window, it can be scrolled using the mousewheel.

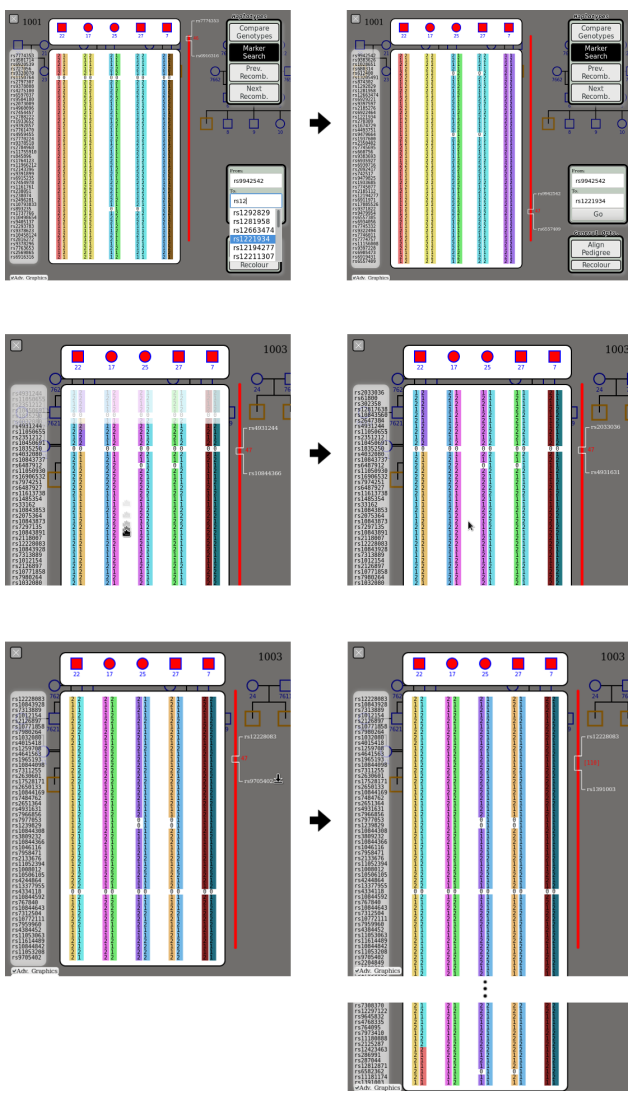

Fig. 4. Modes of interaction: (Top) scrolling the genotypes in the comparison view by dragging and dropping the haplotypes, updating the marker positions in the chromosome overview; (Middle) adjusting the viewport by manually specifying a locus of interest between two markers; (Bottom) resizing the viewport by dragging the upper or lower handle on the chromosome overview, along with a Shift keyboard modifier for permitting slower movement for precision. A viewport larger than the size of the window can be scrolled using the mousewheel, but the default behaviour can be reverted by holding Ctrl and dragging one of the handles to snap the viewport back to the window size.

2.3 IBD Colouring

In HAPLOSCOPER, haplotypes are coloured by IBD by converting the task of resolving ambiguous parentage into a path-finding problem and performing A* search to determine the path with least recombinations. A* search is an efficient path-finding algorithm used in real-time mapping applications (Algfoor *et al.*, 2015). In a connected graph with weighted edges, an optimal path between two nodes is found by minimising the total edge-cost.

A* search is a best-first search algorithm that, in the process of finding the optimal path, maintains a "frontier" of nodes from which the node deemed most likely to be the next intermediate node on the path to the target node is selected. The search procedure is admissible on condition that the estimated cost to the target node is not greater than the true cost from the next intermediate node to the target node (Hart *et al.*, 1968), under the following heuristic: $f(n) = g(n) + h(n)$, where n is an intermediate node on the path, $g(n)$ is the cumulative cost of the path (from start node to n), and $h(n)$ is the heuristic that estimates the lowest cost from n to the target node.

In a genomic context this can be conceptualised as a multi-layer network graph, where edges only exist between consecutive layers. Each layer represents an individual-marker locus, and each node a distinct founder allele (Figure 5). The algorithm traverses from one end of a chromosome to the other under the heuristic of minimizing the number of recombinations. A maximum of $2f$ nodes are possible in each layer, where f is the number of founders. This number is often far smaller due to the manner in which the graph is initialised (see Section 3.2.1).

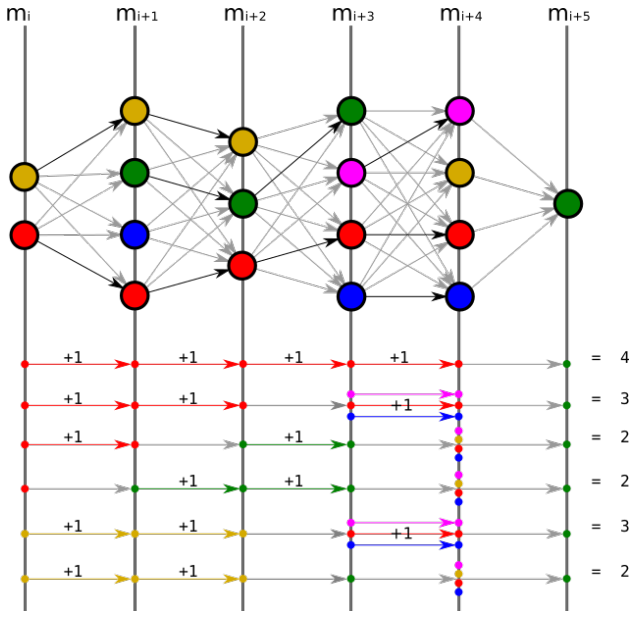

Fig. 5. (Top) A multi-layer network graph depicting five founder alleles as uniquely coloured nodes within a marker locus stretching from m_i to m_{i+5}. Black arrows depict desired contiguous founder-allele stretches, and grey arrows indicate recombinations from one founder-allele to another. (Bottom) Six possible routes explored by the search algorithm, with contiguous stretches being rewarded $+1$ to the total path sum. The first route has the largest path sum of 4 and is the most optimal path in the range considered.

2.4 File Format Support

HAPLOSCOPER was designed to primarily accept phased genotypes from both binary markers (SNPs) and polymorphic markers (e.g. VNTRs and STRs), in vertical or horizontal pre-MAKEPED or modern LINKAGE-based formats with chromosome pairs delineated on adjacent lines.

However, it can additionally utilise supplementary gene flow information output by specific programs. Some applications incorporate this information within the main haplotypes output file, whereas others provide it in a supplementary file. Merlin, for example, outputs founder alleles in a file called `gene.flow` (Abecasis *et al.*, 2002). Allegro and Simwalk both output the optimal descent graph, stating the gene flow between generations (Gudbjartsson *et al.*, 2005; Sobel *et al.*, 2002). The haplotypes from Genehunter and Merlin do not include sex data (Kruglyak *et al.*, 1996), requiring it be inferred through parentage for all but the last generation whose sex is declared unknown. This information can be provided separately. Further marker data (e.g. SNP ID, genetic distance, etc) is displayed upon discovery and preserved across successive sessions in local storage.

3 Methods

3.1 Web Technologies

HAPLOSCOPER is implemented using HTML5 and JavaScript. Unlike other haplotype visualisation programs that require local installation (including installation of dependencies), HAPLOSCOPER runs in any compliant web browser. Features like A* search IBD colouring (described in detail in Section 3.2) make use of JavaScript's typed arrays to eliminate the redundancy of the default numeric float type, compacting large numeric sets into small 8 bit decimal arrays. Fast 2D graphics rendering

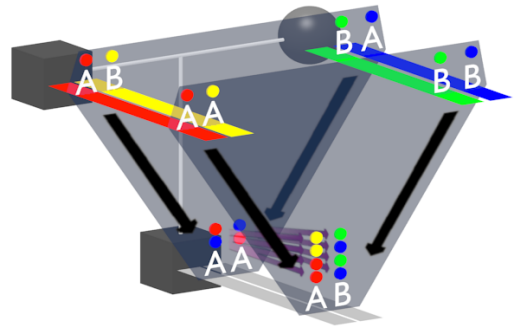

Fig. 6. Parent-Offspring trio for a 2-marker locus. The two vertical layers display the independent haplogroup initialisation occurring at each marker locus, with the proliferation of founder-alleles (coloured dots) from parent-to-offspring under valid genotype configurations. The A* search process then bisects these layers for each non-founder allele, to maximize the length of a contiguous founder-allele.

is performed using the HTML5 canvas-based KineticJS library (`http://kineticjs.com`). KineticJS renders graphics to layers which are implemented as separate canvas elements.

HAPLOSCOPER uses two layers for drawing operations, each acting either as a framebuffer or backbuffer where necessary to limit the number of redraw operations.

Animation is used to transition between the pedigree drawing and haplotype comparison modes. Visual effects can be disabled if, for example, the browser does not support hardware acceleration and performance is degraded. As of writing, Webkit-based browsers (Chromium, Safari) offer better performance than Mozilla's Gecko-based browser (Firefox), but this is subject to change once the multi-process Firefox model (Electrolysis) is fully incorporated (Mozilla, 2016).

3.2 A* search IBD Colouring

The procedure to ensure that haplotypes are coloured appropriately to reflect identity by descent occurs at the level of parent-offspring trios for every non-founder, ordered through a top-down pass of the pedigree under the assertion that offspring haplotypes will not be processed until the corresponding parental haplotypes are fully resolved.

In each trio, the processing is split into two distinct phases: initialising the haplogroups, and determining the optimal path through an A* search.

3.2.1 Haplogroup Initialisation

Founder-alleles are distributed to non-founders through parent-offspring interactions for valid genotype configurations under a predetermined disease model. A *haplogroup* is defined at each non-founder allele to accumulate all the potentially valid founder-allele assignments inherited through these interactions. Ideally, each haplogroup holds a single founder-allele that explicitly defines the path a founder-allele traverses across multiple generations. In general, however, there tend to be multiple valid paths of descent a given founder-allele can traverse and we must consider all possibilities.

Although our implementation of the A* search assumes that genotype data is phased (i.e.: resulting from haplotype reconstruction), the graph initialization procedure does not make any assumptions on which allele is maternal or paternal, relying solely on founder-allele assignments to populate the haplogroups as shown in Figure 6.

begin
  $X \leftarrow$ parental exclusion set of illegal founder-alleles
  frontier $\leftarrow$ set of active paths, initialized to first haplogroup
  complete $\leftarrow$ set of completed paths, initialized as empty
  while *frontier* > 0 **do**
    $p \leftarrow$ first path in *frontier*
    $F \leftarrow$ set of founder-alleles at marker locus $|p|+1$
    for $f \in F$ **do**
      $s \leftarrow$ perform lookahead and count contiguous stretch of f
      if $(s >$ *minStretch*$)$ and $(f \notin X)$ **then**
        $e \leftarrow$ extend path p by length s with founder-allele f
        if $|e| > |$*markers*$|$ **then**
          push e to *complete*
        else
          push e to *frontier*
        end
      end
    end
    sort *frontier* by desc. length and truncate up to *maxNumPaths*
  end
  sort *complete* by desc. number of recombinations
  return first path in *complete*
end

Algorithm 1: A* search upon a single chromosome pre-initialised with a set of potential founder-alleles at each marker locus.

3.2.2 Finding the Optimal Path

During graph initialisation, genotypes were processed vertically, parent-to-offspring, to independently populate non-founder haplogroups for each marker locus along the length of a non-founder chromosome. To find the optimal path, however, the A* search algorithm traverses these haplogroups, from marker to marker, with each haplogroup representing a diverse selection of founder-alleles. The search operates under the heuristic of maximising contiguous stretches (or *paths*) of the same founder-allele across multiple marker loci; i.e minimizing the total number of recombinations. The branching nature of the search requires consideration of multiple paths, each of which expands the frontier of possible haplogroups. Valid paths are restricted to those that meet minimum stretch requirements and fall within accepted founder-allele selections, as outlined in Algorithm 1.

The founder-alleles assigned during graph initialisation do not state whether they are inherited maternally or paternally. To address this, we define a *parental exclusion set* that encapsulates all the founder-alleles present in either maternal or paternal alleles. The algorithm then selects from these sets to determine where the allele originated from. In the case of a consanguineous relationship, each maternal and paternal exclusion set subtracts against the union of both sets to permit the inclusion of shared groups.

A maximum working set of eight examined paths are expanded upon with paths added/removed according to the number of recombinations. Paths with the same number of recombinations are included but discounted from the working set in order to encourage path diversity.

Once an active path reaches the last haplogroup, it is moved into the set of complete paths to later yield the path with the lowest total number of recombinations after finishing the expansion of all active paths.

3.3 Comparative Analysis

The purpose of haplotype visualisation is to help distinguish between loci whose segregation is concordant with the disease trait from those that are not. In such regions there will be a clear distinction in IBD information, defined by the disease model, between affected and unaffected individuals. To give researchers an overview of their data, HAPLOSCOPER defines a locus-specific score based on the following conditions:

1. Across all pedigrees, the *genotypes* for all selected individuals must match.
2. In a given pedigree, the *haplotypes* for selected individuals must match, i.e. both the genotypes and the founder allele designations are the same.

This effectively performs an enhanced mode of identity mapping, where IBS metrics are generated across adjacent pedigrees whilst each still retaining the family-specific IBD context derived from the haplotypes.

The degree of specificity for the matching requirement is established by the type of zygosity requested. Under a heterozygous setting, only a single allele from each individual would need to be matched to generate a score. For homozygous and compound heterozygous, the matching of both alleles of an individual would be required, with the former requiring that allele pairs be identical.

This is defined explicitly in the equation below, where for a given marker-locus m and family f, scores are generated for each of the three zygosity scenarios as follows:

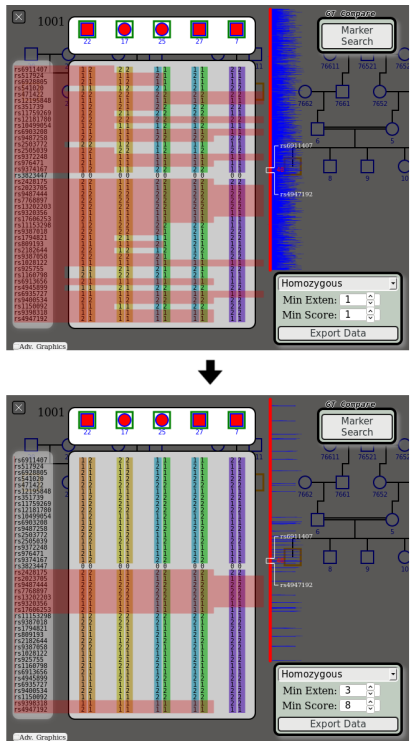

Fig. 7. Comparative view displaying identity scores under a homozygous context, with (Top) no filtering and (Bottom) a minimum score threshold of 8 and a minimum peak size of 3 markers. Scores are overlayed on both the genotypes and the chromosome overview.

$$\text{SCORE}\,(m,f) = \sum_i g(A_i,m) - \sum_j g(U_j,m) \qquad (3.1)$$

where:

A = Affecteds, U = Unaffecteds

A_0 = First selected affected individual

$H_{p,m}$ = Set of haplotypes for individual p at locus m

$$g|het(p,m) = \begin{cases} 1, & \text{if } \exists\ h \in H_{p,m} \ : h \in A_{0,m} \\ 0, & \text{otherwise} \end{cases}$$

$$g|chet(p,m) = \begin{cases} 1, & \text{if } H_{p,m} = A_{0,m} \text{ and } |A_{0,m}| > 1 \\ 0, & \text{otherwise} \end{cases}$$

$$g|hom(p,m) = \begin{cases} 1, & \text{if } H_{p,m} = A_{0,m} \text{ and } |A_{0,m}| = 1 \\ 0, & \text{otherwise} \end{cases}$$

The graphical representation is then the summation of this score across all families, to be overlayed upon both the genotypes and the chromosome overview (Figure 7). Additional refinement can be performed by the user such as specifying a minimum score threshold to filter out less significant peaks, and setting a extension lower-bound to yield broader peaks.

4 Discussion

HAPLOSCOPER provides a unified environment to create, analyse, and visualize pedigrees together with their associated haplotypes. Pedigree creation allows for large families to be drawn using the mouse and exported or saved to local storage between sessions. Processing for IBD colouring is highly efficient and the results viewable across multiple families simultaneously. We provide several methods to inspect the displayed haplotype data including: displaying haplotypes between user-specified flanking markers, skipping between recombination points and scoring by haplotype consistency with the disease model.

4.1 Path-finding approach

The A* search processes each sister chromatid independently of one another, however the correct resolution of one chromatid depends upon the correct resolution of the other due to the mutual-exclusivity of the maternal or paternal exclusion set they are processed against. Due to the parent non-specific means in which HAPLOSCOPER initialises founder-alleles, this may prompt the A* search to reprocess both chromatids with swapped parental exclusion sets if a path cannot be determined initially. Though seemingly costly, the lack of strict phasing during the haplogroup initialisation stage and the serial processing of chromatids provides a greater flexibility in resolving a larger scope of genetic disorders. In the future this could be adapted to explore monosomy, trisomy, and tetrasomy cases.

4.2 Visualisation Accuracy

The haplotype visualisation performed by HAPLOSCOPER was compared with HaploPainter. For all autosomal pedigrees analysed, the same points of recombination were identified from Allegro (ihaplo.out), Genehunter (haplo.chr), Merlin (merlin.chr), and Simwalk (HEF.ALL) output files. Beyond the simple pedigrees, we have used HAPLOSCOPER with four non-trivial families: autosomal dominant (27 members, 23-bit); a highly consanguineous autosomal recessive (24 members, 29-bit); and an X-linked dominant (17 members, 15-bit).

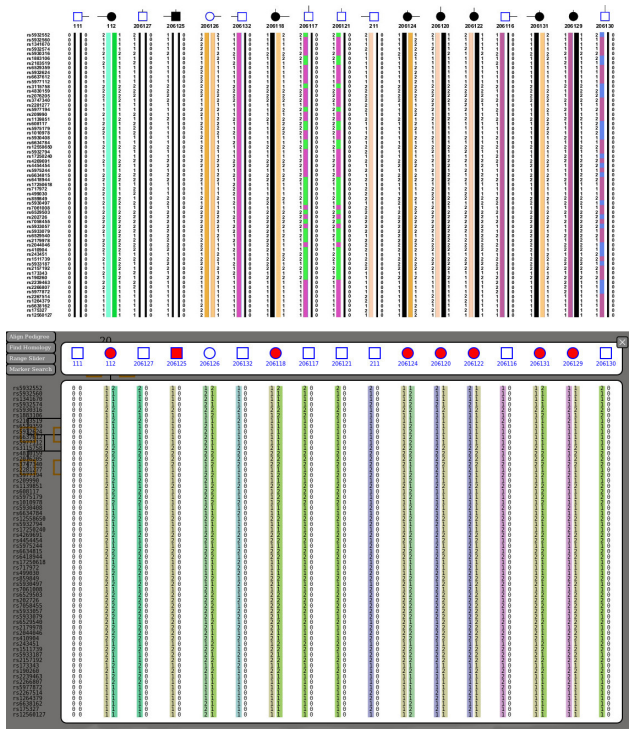

Fig. 8. A comparison of the X-linked dominant pedigree showing a mid-region of chrX spanning 72 markers. (Top) HaploPainter with output modified only for horizontally alignment, and (Bottom) HAPLOSCOPER showing all members via default comparison view.

Fig 8 provides a side-by-side comparison of the same X-linked pedigree shown previously (Fig 1) under a larger marker context, where HAPLOSCOPER shows the correct IBD colouring. HaploPainter normally displays haplotypes inline with the pedigree, but this was modified in Figure 8 for comparison.

4.3 Privacy

HAPLOSCOPER currently operates in-browser either installed locally or through the web, with analyses restricted to a single user. In the interests of scientific collaboration, it is likely that the end-user would want to share their analysis with other researchers working on the same project. Due to the sensitivities of patient data, however, as well as the possibility of identifying individuals based on pedigree structure alone, HAPLOSCOPER was designed with the intention of not requiring any client-server communication after the web application has loaded. The discretion of patient data is ultimately left to the user, and we provide the option to strip patient names and other annotations on export.

4.4 Future Work

HAPLOSCOPER was built on top of KineticJS because of its stability; active development being frozen since 2014. However, in order for HAPLOSCOPER to benefit from performance improvements it will need to migrate to one of the primary alternatives, either ConcreteJS (`http://concretejs.com/`), by the author of KineticJS or KonvaJS (`https://github.com/konvajs/`) that both other distinct features and advantages that will need to be evaluated.

Future versions of HAPLOSCOPER will aim to integrate the visualization and creation modes to provide more flexibility, for example,

to allow for modifying an existing pedigree after haplotype data is loaded. Additional features could include SVG export and selective visualisation of multiple regions to help produce publication quality figures.

Acknowledgements

R.K. is supported by St. Peter's Trust for Kidney, Bladder and Prostate Research, the David and Elaine Potter Charitable Foundation, Kids Kidney Research, Garfield Weston Foundation, Kidney Research UK, the Lowe Syndrome Trust, the Mitchell Charitable Trust, and the European Union, FP7 (grant agreement 2012-305608 "European Consortium for High-Throughput Research in Rare Kidney Diseases (EURenOmics)".

Conflict of interest: None declared.

References

Abecasis, G. R., Cherny, S. S., Cookson, W. O., and Cardon, L. R. (2002). Merlin-rapid analysis of dense genetic maps using sparse gene flow trees. *Nature Genetics*, **30**(1), 97–101.

Algfoor, Z. A., Sunar, M. S., and Kolivand, H. (2015). A comprehensive study on pathfinding techniques for robotics and video games. *International Journal of Computer Games Technology*, **2015**, 7.

Bennett, R. L., Steinhaus, K. A., Uhrich, S. B., O'Sullivan, C. K., Resta, R. G., Lochner-Doyle, D., Markel, D. S., Vincent, V., and Hamanishi, J. (1995). Recommendations for standardized human pedigree nomenclature. *Journal of Genetic Counseling*, **4**(4), 267–279.

Bennett, R. L., French, K. S., Resta, R. G., and Doyle, D. L. (2008). Standardized human pedigree nomenclature: update and assessment of the recommendations of the National Society of Genetic Counselors. *Journal of genetic counseling*, **17**(5), 424–433.

Bockenhauer, D., Medlar, A. J., Ashton, E., Kleta, R., and Lench, N. (2012). Genetic testing in renal disease. *Pediatric Nephrology*, **27**(6), 873–883.

Gudbjartsson, D. F., Thorvaldsson, T., Kong, A., Gunnarsson, G., and Ingolfsdottir, A. (2005). Allegro version 2. *Nat Genet*, **37**(10), 1015–1016.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, **4**(2), 100–107.

Kruglyak, L., Daly, M. J., Reeve-Daly, M. P., and Lander, E. S. (1996). Parametric and nonparametric linkage analysis: a unified multipoint approach. *American Journal of Human Genetics*, **58**(6), 1347–1363.

Mozilla (2016). What's Next for Multi-Process Firefox. `https://blog.mozilla.org/futurereleases/2016/08/02/whats-next-for-multi-process-firefox/`.

Sobel, E., Papp, J. C., and Lange, K. (2002). Detection and Integration of Genotyping Errors in Statistical Genetics. *American Journal of Human Genetics*, **70**(2), 496–508.

Thiele, H. and Nürnberg, P. (2005). HaploPainter: a tool for drawing pedigrees with complex haplotypes. *Bioinformatics*, **21**(8), 1730–1732.