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

Subject Section XXXX

HAPLOTYPE: 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 due to lack of native support.
Results: To resolve these issues, we developed HAPLOTYPE, a novel web application for haplotype visualisation and pedigree drawing. HAPLOTYPE 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. HAPLOTYPE can currently process haplotype reconstruction output from Allegro, Genehunter, Merlin and Simwalk.
Availability: HAPLOTYPE is licensed under GPLv3 and is hosted and maintained via Bitbucket.
Web Application: `http://mtekman.bitbucket.io/haplo_html5`
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

Family studies remain an important approach to investigate the aetiology of monogenic disorders. 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 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 display haplotypes together with the pedigree structure and to colour haplotypes based on identity by descent (IBD).

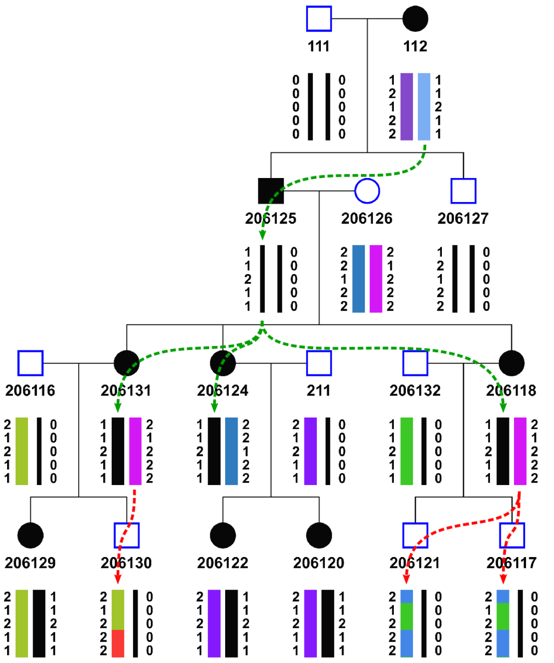


Fig. 1. HaploPainter interpretation of a five marker X-linked analysis. Colours indicate identity by descent. Arrows are overlayed to show the true flow of genetic data based on genotypes, with green showing inconsistent colouring between successive meioses and red showing erroneous inheritance.

There are many programs available for visualization, with HaploPainter being the most popular by number of citations (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 Fig 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).

To resolve these issues we present HAPLOTYPE, a novel web application for haplotype visualization and pedigree drawing. HAPLOTYPE is designed specifically for navigating high numbers of markers, providing a more intuitive viewing mode compared to other programs. The cause of HaploPainter's IBD colouring issues upon the X chromosome was identified as a failure to correctly handle allosomic data due to a lack of contingency checks that permit unrestrained modes of descent. The basis of this oversight stems from the core methodology in which haplotypes are resolved, the scope of which appears to affect the entirety of the inheritance processing portion of their application. Our novel A* search-based method overcomes these pitfalls to ensure that haplotypes are displayed correctly.

Finally, HAPLOTYPE is web-based and therefore runs in any HTML5-compatible 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.

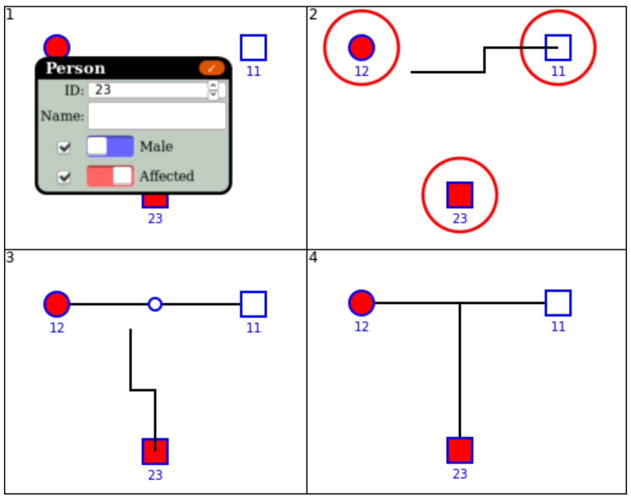

Person
ID: 23
Name:
☑ Male
☑ Affected

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.

2 Approach

HAPLOTYPE 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 via double-lines. Pedigrees can be loaded and saved from local browser storage. Pedigrees can be imported and exported in standard LINKAGE (pre-makeped) format.

2.2 Haplotype Visualisation

HaploPainter, visualises haplotypes in-line with the pedigree. However, for large numbers of markers, comparing the same region of interest between several generations is inconvenient, requiring the user to scroll down and find the same locus over and over again. In our experience, this makes the procedure unnecessarily time-consuming and can lead to errors.

In HAPLOTYPE, haplotype comparison is performed in a separate viewing mode. Selected individuals are aligned vertically and grouped by family. Haplotypes are displayed underneath each individual, allowing

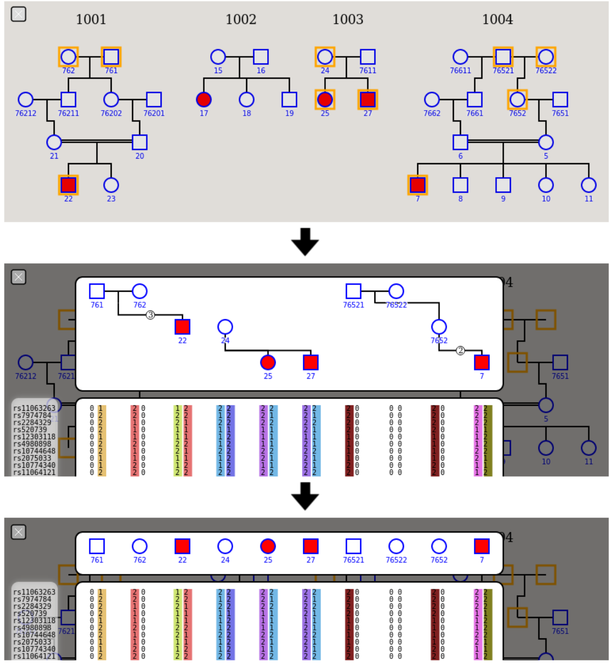


Fig. 3. (Top) SelectionView enabling the subselection of individuals across multiple pedigrees; (Middle) ComparisonView displaying the haplotypes of selected individuals, connected by lines indicating their degrees of separation from one another; (Bottom) ComparisonView representing the same genotypes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

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).

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 upon the chromosome.

The viewport can be manipulated by a variety of different interactions as shown in Fig 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 an interactive 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 Fig 4 (Bottom). The size of the viewport is locked to the size of the window by default, but upon setting a larger region will allow the window scrolling.

2.3 IBD Colouring

In HAPLOTYPE, 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.

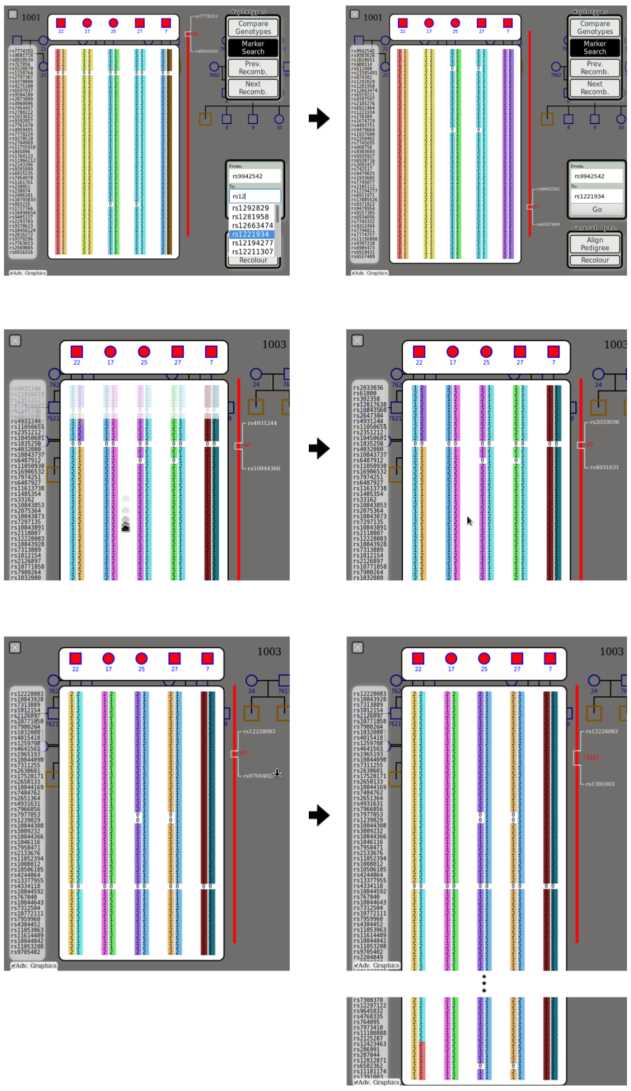


Fig. 4. (Top) Scrolling the genotypes in ComparisonView by dragging and dropping the haplotypes which update the marker positions in the chromosome overview. (Middle) Adjusting the viewport by manually specifying a locus of interest between two markers. (Bottom) Resizing the viewport by dragging either the upper or lower handle on the chromosome overview, with a Shift keyboard modifier for permitting slower movement for precision. A viewport larger than the size of the window will prompt window scrolling upon mousewheel input, 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.

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

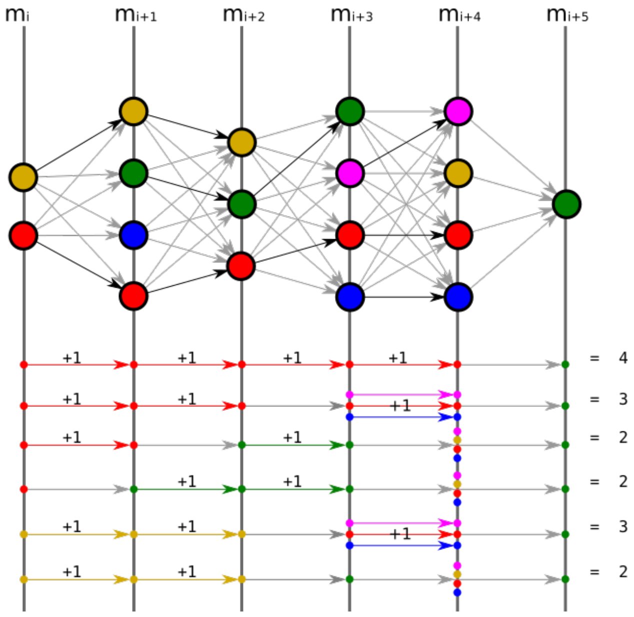


Fig. 5. (Top) A multi-layer network graph depicting five founder alleles as uniquely coloured nodes within a marker locus stretching from m_i to m_{i+5}. Black arrows depict desired contiguous founder group stretches, and grey arrows indicate recombinations from one founder group to another. (Bottom) 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 region under consideration.

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).

2.4 File Format Support

HAPLOTYPE was designed to primarily accept phased genotypes in vertical or horizontal pre-MAKEPED or modern LINKAGE-based derived formats with sister homologs 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 (Abecasis *et al.*, 2002), for example, outputs founder alleles in a file called `gene.flow`. Allegro (Gudbjartsson *et al.*, 2005) and Simwalk (Sobel *et al.*, 2002) both output the optimal descent graph, stating the gene flow between generations. Haplotypes from Genehunter (Kruglyak *et al.*, 1996) and Merlin do not include sex data, 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

HAPLOTYPE is implemented using HTML5 and JavaScript. Unlike other haplotype visualisation programs that require local installation (including installation of dependencies), HAPLOTYPE 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 is performed using the HTML5 canvas-based KineticJS library (`http://kineticjs.com`). KineticJS renders graphics to layers which are implemented as separate canvas elements.

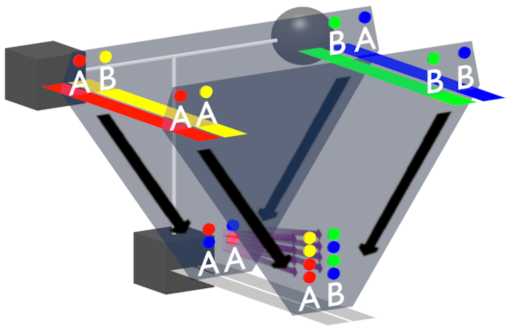


Fig. 6. Parent-Offspring trio for a 2-marker locus. The two vertical layers display the independent graph initialisation occurring at each marker locus across all individuals, with the proliferation of founder-groups (coloured dots) from parent to offspring for valid genotypes configurations. The A*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

HAPLOTYPE 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 is split into three distinct phases: initialising the network graph, determining the optimal path and final cleanup operations.

3.2.1 Graph Initialisation

We initialise the graph with a top-down pass of each pedigree to create founder allele nodes and, for each non-founder, define the set of alleles that can be inherited from each parent at each locus. This can be viewed as a vertical bisection of the genotype data, where for m genetic markers there would be m pedigrees of duplicate structure but differing allelelic content that would represent the network graph of a particular marker-locus as shown in Fig 6.

Each graph is populated with nodes at every individual connected by edges representing the parent-offspring relations depicted by the original pedigree. Founders propagate their alleles throughout the graph, but due to the limited polymorphism offered by genetic markers it is difficult to distinguish between IBS and IBD states in non-founders. Resolving this ambiguity is the crux of the processing, and it is aided by concurrently assigning all founder alleles to their own *founder-group*. These founder

Table 1. Trio allele configurations, with applicable Mendelian child genotypes describing four levels of inheritance ambiguity. An allele is Allele-Specific (AS) if it can unambiguously match a specific allele from a given parent, and Parent-Specific (PS) if a specific parent can be matched.

Parental Alleles		*Child Alleles*			
Case	*GTs*	*Mend.*	*AS*	*PS*	*Group*
Homozygous	AA ǀ AA	AA	××	××	Orphaned
	BB ǀ BB	BB	××	××	Orphaned
	AA ǀ BB	AB	××	✓✓	Bound-Parent
Heterozygous	AB ǀ AB	AA	✓✓	××	Bound-Allele
		BB	✓✓	××	Bound-Allele
		AB	✓✓	××	Bound-Allele
Homozygous with Heterozygous	AA ǀ AB	AA	××	××	Orphaned
		AB	×✓	×✓	Locked
	BB ǀ AB	BB	××	××	Orphaned
		AB	✓×	✓×	Locked

groups are transmitted via parent-offspring relations into non-founder *haplogroups*, which each contain the founder-group assignment that describes the allele.

Ideally, each haplogroup holds a single founder-group that explicitly logs the path a founder allele traverses across multiple generations. In reality, there are multiple valid paths of descent a single founder allele can travel and we must consider each and every one of them.

For each valid parent-offspring genotype configuration detected, the corresponding parental haplogroups are appended to the child haplogroups which are in turn passed on until the final generation is reached. The number of potential founder-groups held within a haplogroup grows quadratically with each generation up to a maximum of $2f$ groups for f founders.

Though the A* algorithm makes use of phased genotype data, the graph initialization procedure does not make any assumptions upon which allele is maternal or paternal, relying solely on founder-group assignments to populate the haplogroups.

The complexity of resolving parental alleles with child alleles can be summarised by four tiers of allele-pair specificity (in ascending order of complexity): Locked, Bound-Parent, Bound-Allele, and Orphaned. Locked alleles can unambiguously map at least one allele to a specific parent and to a specific allele within that parent. Bound-Parent alleles can match both alleles to a parent, but not to a specific allele within that parent. Bound-Allele is the opposite; where both alleles can be assigned to either allele in the parent, but the parent is non-specific. Orphaned provides no specificity. A summary of valid genotype configurations and their specificity is outlined in Table 1. In X-linked analysis, the process is adapted to reflect the fact males are hemizygous and therefore have only a single allele.

3.2.2 Finding the Optimal Path

Where graph initialisation processed genetic data in a vertical generational context to create, here we traverse horizontally through the populated haplogroups for each non-founder allele as a multi-layer network graph.

The A* search algorithm then attempts to find the optimal path through the graph, with the aim of maximising contiguous stretches (or *paths*) of the same founder-group across multiple layers; i.e minimizing the total number of recombinations. The branching nature of the search prompts the consideration of multiple paths which each expand a frontier of haplogroups. Valid paths are restricted to those that meet minimum stretch requirements and fall within accepted founder-group selections, as outlined in Algorithm 1.

Due to the non parental-specificity of the potential founder-groups assigned during the graph initialization process, the allele being traversed can be either maternally or paternally inherited. To address this, we define a *parental exclusion set* that encapsulates all the founder-groups present in either maternal or paternal alleles. The algorithm then selects against these sets across alternate runs upon the same allele in order 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.

```
begin
    prefetch relevant chromosome
    maxPath ← global max. num. of paths to explore
    numMark ← total num. markers in chromosome
    P ← parental exclusion set of illegal colours
    complete ← initialise empty list of completed paths.
    frontier ← array of working paths, initialised with array of
    colours at first marker-layer in chromosome as starting points
    while frontier > 0 do
        sort frontier by desc. length and select first maxPath
        a ← shift frontier to select first active path
        C ← colours in path a at marker-layer length a - 1
        for c ∈ C do
            s ← perform lookahead and count contiguous stretch of c
            if s < 1 or c ∈ P then skip c
            r ← clone path a with c appended s times
            if length r > numMark - 1 then
                push r to complete
            else
                push r to frontier
            end
        end
    end
    sort complete by desc. length
    return shift complete
end
```

Algorithm 1: A* search upon a chromosome of pre-initialised multi-layer network graph, with founder alleles represented as colours. Sort operations apply in-place, and shift operations truncate from the head of an array.

A maximum working set of eight examined paths are expanded upon with paths added/removed as determined by their respective running totals upon each iteration of the working set. Paths with the same running total of recombinations are included but discounted from the working set in order to encourage more path diversity.

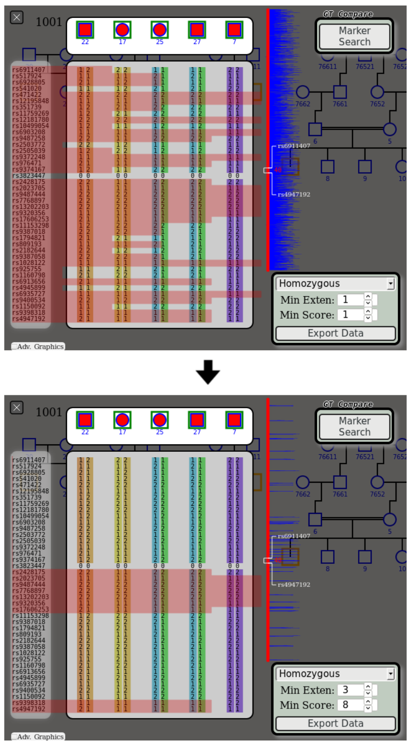
Once an active path reaches the last layer of founder-groups, it is moved into the set of complete paths to later yield the path with the lowest total number of recombinations after finishing the expansion of all active paths.

3.2.3 Cleanup Operations

Upon full evaluation of a pedigree, the network graphs associated with each marker locus are discarded and only the optimal paths are stored.

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, HAPLOTYPE defines a score based on the disease model that is plotted on the chromosome overview to identify such regions. To give an example of how this works, assuming a dominant disease trait, if at a given locus all affected individuals have



Homozygous
Min Exten: 1
Min Score: 1
Export Data

Homozygous
Min Exten: 3
Min Score: 8
Export Data

Fig. 7. HomologyView displaying homology 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 upon both the genotypes and the chromosome overview.

either $\{red, red\}$ or $\{red, green\}$ founder alleles and the unaffected individuals have $\{green, green\}$, then this region will receive a high score. If, on the other hand, the model were recessive, it would receive a lower score. The score is additive across pedigrees under the assumption of genetic homogeneity. Scores can be exported to a text file.

For a given marker-locus *m* and family *f*, scores are generated for each of the three zygosity scenarios as follows:

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

where:

A = Affecteds, $\quad U$ = 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 a summation of this score across all families, to be overlayed upon both the genotypes and the chromosome overview (see Fig 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

HAPLOTYPE 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 is not restricted to SNPs, but can accept polymorphic markers (e.g. VNTRs and STRs). The path-finding approach processes each chromosome separately with the only interaction between sister homologs emerging from the mutually-exclusive parental exclusion sets, such that the use of one set on one allele cannot then be used upon the other. In the future this could be adapted to explore monosomy, trisomy, and tetrasomy cases.

4.2 Visualisation Accuracy

The haplotype visualisation performed by HAPLOTYPE 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 HAPLOTYPE 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 provides a side-by-side comparison of the same X-linked pedigree shown previously (Fig 1) under a larger marker context, where HAPLOTYPE shows the correct IBD colouring. HaploPainter normally displays haplotypes in a generational view, but this was modified in Fig 8 for better comparison.

4.3 Privacy

HAPLOTYPE currently operates in-browser via either local or web deployment, and analyses are 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, HAPLOTYPE 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

HAPLOTYPE was built on top of KineticJS because of its stability; active development being frozen since 2014. However, in order for HAPLOTYPE 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.

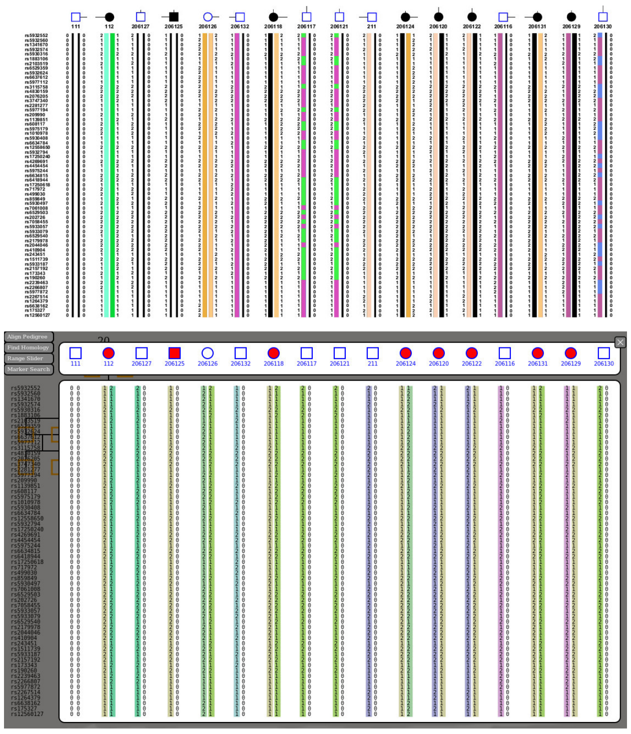


Fig. 8. A comparison of the X-linked dominant pedigree showing a mid-region of chrX spanning 72 markers. (Top) HaploPainter with output modified only for horizontally alignment, and (Bottom) HAPLOTYPE showing all members via default Comparison View.

Future versions of HAPLOTYPE 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.