המכללה האקדמית להנדסה
אורט בראודה
ORT BRAUDE COLLEGE

Software Engineering Department

Ort Braude College

Genetic Mapping in The Case of Huge Number of Markers

Authors:

Alaa Grable

Adam Mahameed

Supervisor:

Dr. Zeev Frenkel

Content

1. Introduction 3
2. Scientific Background & Related Work 4
2.1 Some biological background 4
2.2 DNA, chromosomes, genes 5
2.3 Genetic markers 6
2.4 Genetic Recombination 7
2.4.1 Recombination and mapping models 7
2.5 Genetic mapping 8
2.6 Design of mapping population 9
2.7 Genetic map construction based on genotypes 10
2.7.1 Data filtration 10
2.7.2 Grouping markers onto groups of closely related markers 11
2.7.3 Estimation of recombination rates 11
2.8 Cluster analysis 11
2.8.1 Single-linkage clustering 12
2.8.2 K-Means Clustering 12
2.9 Ordering of skeleton markers within linkage groups 13
2.9.1 Travelling salesman problem 13
2.9.2 Minimum Spanning Tree 14
2.9.3 The Branch and Bounds 14
2.10 Genetic algorithm 15
2.11 Short review of existing packages 15
3. Algorithm 16
3.1 Data format: 16
3.1.1. Genotypes: 16
3.1.2. Genetic map 16
3.2 Input and output of our program 16
3.3 Algorithm for building genetic map 17
4. Goals 18
5. Preliminary Software Engineering Documentation 18
5.1. Requirements (Use-Case) 18
5.2 Class diagram 21
5.3 GUI Design 22
5.4 Testing 30
6. Results & Conclusions 31
References 34

Abstract. *The ordering of genetic markers along the genome is a very important resource for the biologists. In particular, it is used in searching for genomic regions and genes affecting various traits of interest. In classical formulation, such ordering is a partial case of Travel Salesman Problem (TSP) searching the shortest path on the net, where nodes are markers and edge lengths are estimated genetic distances. In modern genetic mapping projects the number of genetic markers is huge. Indeed, most of the markers are not different by recombination within a not too large population with a small number of generations. Hence, accuracy of the best possible genetic map is restricted. Most existing software for genetic mapping are working well only for small numbers of markers or for high quality data. In our project we build an effective tool to deal with situations where a number of markers is huge, genotypes are obtained with some amount of non-symmetric errors, and a lot of information is missed. For these purposes we will use various approaches for clustering including single linkage, k-means and network based. Ordering of markers will be based on draft ordering that will be resulted from Minimal Spanning Tree (MST) for reliable clusters. In addition to a genetic map in a standard format (where ordering is fixed) we generate a map in mathematically more correct (but more complicated for biologists) format where non-resolved local orderings are indicated. We hope that the resulting tool will be more powerful than existing ones (freely available or commercial, e.g., MST, Concorde, MultiPoint etc.).*

Key words: *Genetic mapping, Data mining, TSP, Missing data, Data with errors.*

1. INTRODUCTION

Development and life of the organisms are highly dependent on genetic material that is considered to be equal in all cells of the organism. Genetic material (genome) is organized into chromosomes with one-dimensional topology structure. Specific regions of the chromosomes (genes) are responsive for protein production. For many genes and intra-genetic regions, function is unknown. One of the ways to understand what genomic regions are responsive for a specific function is to localize regions associated with this function. For this purpose, biologists use specific genetic markers distributed along the chromosomes. Ordering of markers along the chromosomes enables correct union of noised information from various sources. So, the ordering of genetic markers along the genome is a very important resource for the biologists. There are various approaches for such ordering. In our project we deal with ordering based on frequencies of observed recombination events (genetic mapping). Recombination events are considered as independent where probability of an odd number of recombination's monotonically increases with increasing physical distance between two points on the chromosome. Hence, low recombination rate points on relatively close physical distance on the chromosome. In modern genetic mapping projects, the number of genetic markers is huge. Indeed, most of the markers are not different by recombination within a not too large population with a small number of generations. Hence, accuracy of the best possible genetic map for not too large mapping population is restricted.

Various software packages for genetic mapping were developed. Each of them has some shortcomings. For example, older packages enable working with relatively small amounts of markers. Even modern software enables robust reliable results only in the cases of high-quality data or relatively small number of markers. In our project we build an effective tool to deal with situations where a number of markers is huge, genotypes are obtained with some amount of non-symmetric errors, and a lot of information is missed. For these purposes we will use various approaches of data mining including clustering, network topology study, data imputation and correction, linear structure recognition, statistical analysis, etc. The algorithms and GUI will be implemented with Python. This will help to maintain flexibility of the software and will enable cross platform usage. In addition to genetic maps in a standard format (where ordering is fixed) we generate a map in mathematically more correct (but more complicated for biologists) format where non-resolved local orderings are indicated. We will mainly deal with the estimated observed recombination rates and find genetic coordinates such that observed recombination rates will be similar to ones estimated based on the genetic map. The input data for our program is genotypes (allele states for markers for all individuals of mapping population). The output is the genetic map and characteristics of the quality. We hope that the resulting tool will be more powerful than existing ones.

2. SCIENTIFIC BACKGROUND & RELATED WORK

2.1 Some biological background

Usually we deal with complex organisms or plants consisting of many parts named cells. Generally, we consider some simplified model that all cells of the organisms are originated by mitotic division from a single cell obtained by fusion of cells (like egg and sperm) bringing genetic material from two parental organisms. Development and life of the organisms are highly dependent on genetic material that is considered to be equal in all cells of the organism. Genetic material is mostly organised onto very long molecule DNA subdivided on several parts named chromosomes (see below). Specific protein coding segments of the chromosomes are named by genes. Differable variants in genes and in other specific segments of the chromosome are named by alleles. DNA segments with readable allele state are named by genetic markers. Usually, organisms have two copies of each chromosome (one inherited from one parent and one from the other).

With some simplification, chromosomes are considered as having one-dimensional (linear) structure. Such structure enables genetic recombination that produces new combinations of alleles. Generally, recombination events are random. The probability of an odd number of recombinations monotonically increases with increasing physical distance between two points on the chromosome. This means that allele states in closely situated markers are correlated and associations between alleles can be used for estimation of distances between markers on the genome. These distances can be used for estimation of ordering of markers within the chromosomes. Such ordering with characteristics of recombination frequencies are named genetic maps. For simplicity, to characterize recombination frequencies (rates) genetic coordinates are used. For example, if markers m_1 and m_2 have coordinates $c(m_1)$ and $c(m_2)$ then genetic distance between them is $d=|c(m_2)-c(m_1)|$ and the recombination rate is $r=f(d)$. Function $f(d)$ is defined by the model of recombination events dependence. In the simplest case, recombination events are considered as independent. Recombination distance is scored in centiMorgans (cM) and $f(d)=0.5(1\text{-}\exp\{-2*0.01*d\})$. 1 cM corresponds to 1% recombination rate [1]. Hence, to build genetic map we need to (i) build some mapping population where associations between alleles are presumably significantly high even for markers that are not too close, (ii) select informative genetic markers, (iii) score genotypes, (iv) estimate observed recombination rates and find genetic coordinates such that observed recombination rates will be similar to ones estimated based on the genetic map. Our project deals with stage (iv). The input for the program that we build are the genotypes (allele states for markers for all individuals of mapping population). The output is the genetic map and characteristics of the quality. Now we describe mentioned terms some more.

2.2 DNA, chromosomes, genes

DNA or Deoxyribonucleic acid is a string of complex long chains of molecules called nucleotides. Those nucleotides coil around each other to form a double helix carrying genetic instructions for the development, functioning, growth and reproduction of all known organisms and many viruses. It is situated in the nucleus of cells. There are only four nucleotides that are ever used, these are Adenine (A), Thymidine (T), Guanine (G), and Cytosine (C). A strand of DNA is much like an extremely long sentence that uses only four letters. This two-strand system is the key to how DNA is able to make copies of itself. This can happen because one strand is 'complementary' to another strand. A always matches up with T and G always matches up with C. Because they pair up, they are called "base pairs" [2].

DNA forms a molecular instruction "book", organized into "paragraphs" (genes) and "chapters" (chromosomes) (Fig. 1). Such logical structure is referred to as a genome. Genomes of organisms of the same species are usually organized in the same way. Local differences make organisms unique genetically. Human genome normally consists of 46 chromosomes (two sets of 23; 23 from the mother and 23 from the father) [3]. Genes are responsible for coding specific proteins. Human genetic code contains about 23,000 genes [4]. Proteins have lots of different roles. They form the scaffolding of cells, as well as helping them to function and communicate. If these genes go wrong, they can cause cells to grow out of control (like in cancer) [5].

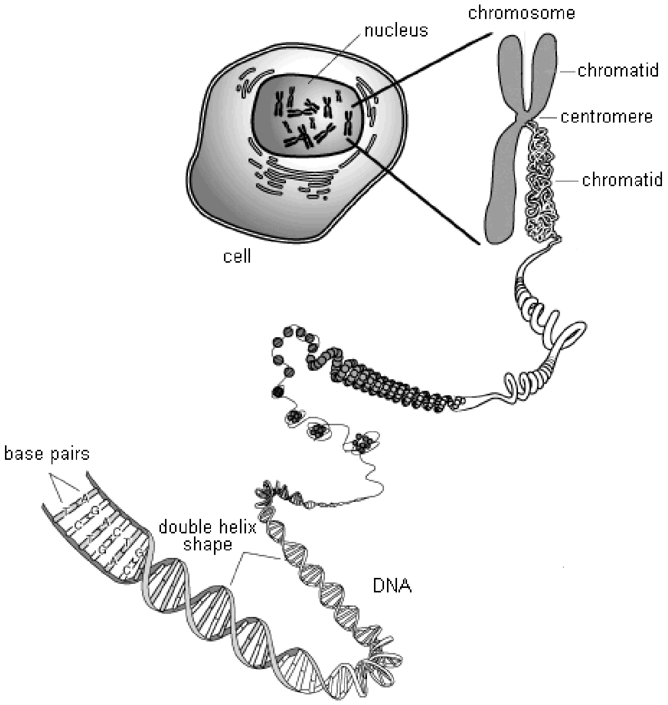


Figure 1. DNA in cell nucleus: chromosome, gene and double helix as model of DNA structure [6]

2.3 Genetic markers

A genetic marker is a part of DNA with some variation among organisms of the species. Variants in such parts are named alleles. Good genetic markers have a unique location in the genome. Such position of marker is named locus. Variations in markers are generally the result of mutation many generations ago. A genetic marker may be a short DNA sequence, such as a sequence surrounding a single base-pair change (single nucleotide polymorphism, SNP), or a long one, like minisatellites (tract of repetitive DNA in which certain DNA motifs are typically repeated 5-50 times) [7]. Alleles in some loci can be scored without knowing DNA sequence in corresponding genome region. For example, it can be based on the morphology of the organism (e.g., eye color), resistance to diseases and stresses, behavior etc. (Fig. 2).

Genetic markers can be used to study the relationship between an inherited disease and its genetic cause (for example, a particular mutation of a gene that results in a defective protein). It is known that pieces of DNA that lie near each other on a chromosome tend to be inherited together. This property enables the use of a marker to determine the precise inheritance pattern of the gene that has not yet been exactly localized [7]. Genetic markers are also employed in genealogical DNA testing for genetic genealogy to determine genetic distance between individuals or populations. Uniparental markers (on mitochondrial or Y chromosomal DNA) are studied for assessing maternal or paternal lineages. Autosomal markers are used for all ancestry [7].

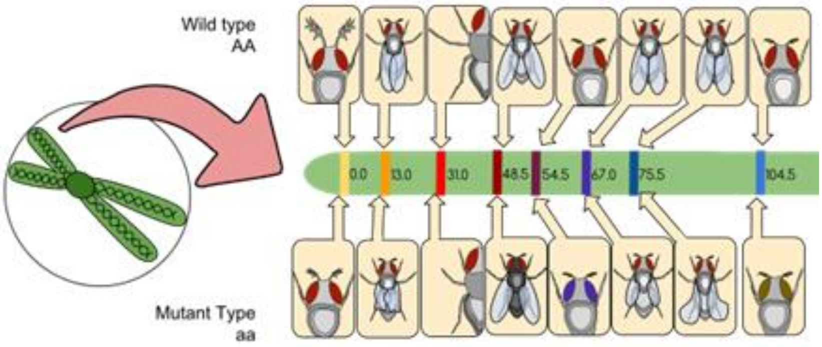


Figure 2. Markers defining morphology of drosophila and alleles that can be scored without sequencing. Numbers reflect their position on the genetic map(see 2.5) [8].

2.4 Genetic Recombination

Genetic recombination (also known as genetic reshuffling) is the exchange of genetic material between different chromosomes which leads to production of offspring with combinations of traits that differ from those found in either parent. In eukaryotes, genetic recombination during meiosis can lead to a novel set of genetic information that can be passed on from the parents to the offspring (Fig. 3).

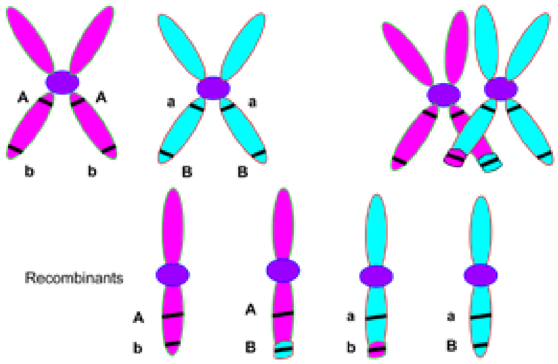


Figure 3. Chromosomal crossover, or crossing over, is the exchange of genetic material between two homologous chromosomes non-sister chromatids that results in recombinant chromosomes during sexual reproduction.

DNA regions that are close together on a chromosome tend to be inherited together during the meiosis phase of sexual reproduction; this is known as genetic linkage. Two genetic markers that are physically near to each other are unlikely to be separated into different chromatids (copy of a chromosome) during chromosomal crossover, and are therefore said to be more linked than markers that are far apart. The distance between two loci is measured in units known as centimorgan (cM). A centimorgan is a distance between loci for which one product of meiosis in one hundred is recombinant. The further two genes are from each other, the more likely they are going to recombine. If it were closer, the opposite would occur.

2.4.1 Recombination and mapping models

Recombination can be considered as random events. Probability r to observe recombination between two markers m_1 and m_2 is defined by distance between these markers $d(m_1,m_2)$ on some scale. Such a scale is named a genetic map. The probability r to observe recombination is named recombination rate or recombination frequency. Indeed, observed recombination between two markers means that the real number of crossing over events between these two markers is odd. Observing no recombinations means that the real number of crossing overs is even. Inheriting markers from different chromosomes are independent. Hence, expected recombination rate between these markers is $r = 0.5$. Markers situated in the same genomic point are always inherited together and recombination rates for these markers $r = 0$. Markers situated too far one from another are also inherited about independently ($r = 0.5$). Hence, function $r(d)$ for calculation of recombination frequency based on distance d should have the following properties: (i) $r(d) \geqslant 0$ for all $d \geqslant 0$, (ii) $r(d)$ is monotonically increased with increasing of d, (iii) $r(0)=0$, $r(d)$ does to 0.5 with d goes infinity.

For markers m_1, m_2 and m_3 consequently situated on the chromosome recombination rate between markers m_1 and m_3 is equal to sum of probabilities that recombination will be observed in the interval $[m_1,m_2]$ but not on $[m_2, m_3]$ and of that recombination will be observed in the interval $[m_2,m_3]$ but not on $[m_1, m_2]$. If recombination rates are independent then $r_{13}=r_{12}(1-r_{23})+(1-r_{12})r_{23}=r_{12}+r_{23}-2r_{12}r_{23}$, where r_{12} and r_{23} are recombination rates between markers m_1 and m_2 and between m_2 and m_3, respectively. This means that $\lim_{\Delta d \to 0} r(\Delta d)/\Delta d=1$ and that $r(d_1+d_2)=r(d_1)(1-r(d_2))+(1-r(d_1))r(d_2)$. Supposing that function $r(d)$ is differentiable, one can write formula for its derivative: $r'(d)=\lim_{\Delta d \to 0} (r(d+\Delta d)-r(d))/\Delta d =\lim_{\Delta d \to 0} (r(d)(1-r(\Delta d))+(1-r(d))r(\Delta d)-r(d))/\Delta d =\lim_{\Delta d \to 0} (r(\Delta d)(1-2r(d)))/\Delta d =(1-2r(d))$ [9 J.B.S.Haldane (1919)]. Solving the linear differential equation $r'(d)=1-2r(d)$ with initial point $r(0)=0$ leads to the following formula: $r(d)=0.5(1-\exp\{-2d\})$. According to this model (named Haldane model) crossovers occur as a Poisson process [10]. If distance is scored in centiMorgans (cM) then 1 cM is corresponding to r =1%, then the formula should be rescaled: $r(d)=0.5(1-\exp\{-2*0.01*d\})$. This formula can be used to estimate genetic distance based on recombination rate: $d= -50 \ln\{1-2r\}$ [11].

Biologically, real crossover events are not independent. In particular, recombination in some point reduces probabilities of recombination in vicinity of this point. Such a situation is better described by Kosambis model: $r(d+\Delta d) \sim r(d)+r(\Delta d)-4r^2(d)r(\Delta d)$ [9]. This leads to equation $r'(d)=1-4r^2(d)$ with solution $r = \frac{(e^{4d/100}-1)}{2(e^{-4d/100}+1)}$, d $= 100 * (1/4 * ln(\frac{1+2r}{1-2r})[cM]$[9]. Indeed, Kosambis model is not biologically based. This is only the simple formula where proportion of double double crossovers is decreased relative to independent recombination cases and such effect is high for short intervals and low for long ones. In our project we will have an option to select what model of recombination to use, Haldane or Kosambi. We do not use other models for mapping functions (reviewed in [12]).

2.5 Genetic mapping

Genetic map is the way to represent relative position of loci on the genome and characterize recombination rates. This means that based on all observed recombination rates reconstructs subdivision of markers on chromosomes and ordering of markers along the chromosomes. In good situations (no missing data, no errors, recombination rates are calculated accurately, density of markers is high enough, no biological problems caused by selection, chromosome distortion etc.) recombination rates between markers from different chromosomes is high. This enables subdividing markers on chromosomes by a single linkage clustering algorithm [13] (Fig. 4). Distances on the recombination map should be corresponded to the observed recombination rates. Such correspondence should be according to the selected mapping function (in our project, Haldane or Kosambi, see above). Setting of markers in incorrect order leads to increasing of the sum of distances between consequent markers. Hence, the ordering of markers can be searched as one corresponding to minimal sum of recombination distances between consequent markers. If we represent the data in the form of a graph with nodes corresponding to markers and edge lengths corresponding to genetic distances then such criteria means that we search for the shortest path on graph edges visiting each node exactly once. This is a form of classical Travel Salesman Problem (TSP) [14].

In practice, data usually has difficulties. Due to technical difficulties, a high percentage of genotyping data is missed and some genotypes are scored with errors. Percentage of missing data and errors can be different for different markers, different genomic regions and different individuals. Some markers have pseudo linkage that can be caused by selection [15]. Pseudo linkage can be also caused by chromosome distortion [16]. Some markers have a dominant effect that can result in significantly non-equal segregation [7]. Additional difficulties arise from unexpected polymorphism in breeding lines, multiple alleles, DNA contamination etc [17]. Large amounts of markers make it impossible to find globally optimal ordering in general cases because TSP is NP-hard [14]. Restricted sizes of mapping population make estimation of recombination rates uncertain. It is especially problematic in cases of dependency of recombination events (genetic interference) [18].

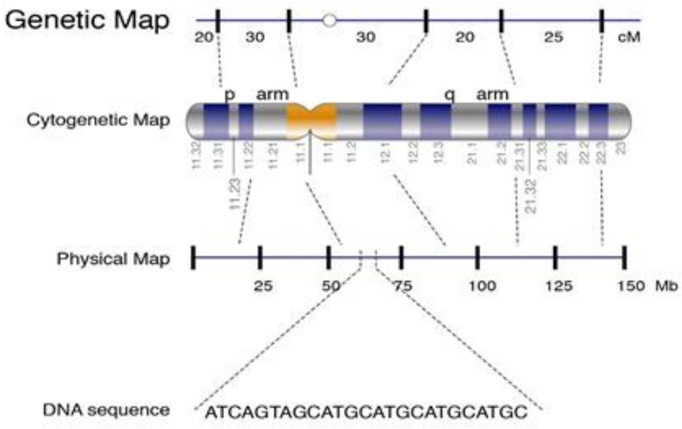


Figure 4. Relations between DNA sequence, physical and genetic maps

2.6 Design of mapping population

Scoring recombination rates is not trivial. For this purpose, we need to know allele phases in parents and progeny. For this purpose, it is reasonable to score recombination rates in specifically designed mapping populations. In particular, to have polymorphism and to know phases of alleles in parents it is much cheaper to use different inbred lines for grandparents [19]. In practice, various mapping population designs can be applied. Here we shortly mention the most popular experiment designs applied for plants and animals where crossing can be planned. In a genetic cross, the parental generation (P) is the first set of parents crossed.

(i) The **F1** (first filial) generation consists of all the hybrids (offspring from the parents).

(ii) The **F2** (second filial) generation consists of the offspring from allowing the F1 individuals to interbreed. So the F2 generation is conventionally produced by the random union of the F1 gametes.

(iii) **Recombinant inbred lines (RILs)** are a collection of strains that can be used to map quantitative trait loci. Parent strains are crossed to create recombinants that are then inbred to isogenicity, resulting in a permanent resource for trait mapping and analysis. Here we describe the process of designing and constructing RILs. This consists of the following steps. Parent strains are selected based on phenotype, marker availability, and compatibility, and they may be genetically engineered to remove unwanted variation or to introduce reporters. A construction design scheme is determined, including the target population size, if and how advanced intercrossing will be done, and the number of generations of inbreeding. Parent crosses and F1 crosses are performed to create an F2 population. Depending on design, advanced intercrossing may be implemented to increase mapping resolution through the accumulation of additional meiotic crossover events. Finally, lines are inbred to create genetically stable recombinant lines

(iv) **Backcross**, the mating of a hybrid organism (F1, offspring of genetically unlike parents P) with one of its parents or with an organism genetically similar to the parent. The backcross is useful in genetics studies for isolating (separating out) certain characteristics in a related group of animals or plants. Basically, backcross enables breeders to transfer a desired trait such as a transgene from one variety (donor parent, DP) into the favored genetic background of another (recurrent parent, RP) [20]. The goal of backcrossing is to obtain a line as identical as possible to the recurrent parent with the addition of the gene of interest that has been added through breeding. In genetic mapping, backcross enables dealing with diploids with tools developed for haploids. In markers monomorph in parental populations backcross offspring carry homozygote on recipient allele or heterozygous (with probabilities 0.5:0.5). In our project we consider only the case of backcross based mapping population.

2.7 Genetic map construction based on genotypes

Classical approach for genetic map construction includes the following stages: (i) data filtration, (ii) grouping markers onto groups of closely related markers, (iii) estimation of recombination rates, (iv) clustering of markers onto linkage groups, (v) selection of skeleton markers, (vi) ordering of skeleton markers, (vii) mapping of other markers, (viii) merging and editing of genetic maps for linkage groups [21].

2.7.1 Data filtration

To reduce the effect of putative errors the first stage is the filtering out presumably problematic data. The standard criteria for filtration of markers are:

(i) Relatively high proportion of missing data. Such markers are less informative. With less observations we have less chance to observe rare recombination events. This leads to underestimation of recombination rates. Smaller recombination rates are falsely interpreted as genetic linkage. False linkage results in difficulties in clustering and ordering of markers.

(ii) Significantly high deviation from equal allele segregation. In the considered experiment design (backcross) we expect that alleles inherited from the hybrid have equal probabilities to be from parental line 1 or line 2. Hence, expected allele frequencies inherited from the hybrid should be p=0.5 and q=1-p=0.5 and expected variance V=p(1-p)/n=0.25/n, where n is the size of the mapping population. This means what with probability 0.99 p should be in the interval [0.5-3*0.5/sqr(n), 0.5+3*0.5/sqr(n)]. Alleles at markers with p highly deviated from 0.5 are questionable. Such situations can be caused by the effect of dominance or unexpected selection. False allele calls and selection lead to underestimation of recombination rates and to false linkage.

(iii) Markers with genotypes too similar to others. Markers with too similar genotypes are presumably situated in the close positions on the genome. Each additional equal genotype provides no additional information for ordering of distant markers. Hence, calculations can be simplified by excluding extra copies of data from the consideration. It is enough to have only one copy of genotype and others will be mapped on to the same position.

(iv) Markers with genotype dissimilar to all others (in the case of high density of markers). In the case of high marker density we expect that each marker has linked neighbours with similar genotypes. Absence of such neighbours can point to putative errors in genotypes. Such errors lead to overestimation of recombination rate and false missing of linkages.

(v) Markers with unexpected topology of a network of tight linkages. From the classical genetics we expect that chromosomes usually have one-dimensional (linear) structure. If the topological structure of a network of linkages is contradicting a one-dimensional chromosome structure then it points that some of such linkages are presumably false (Fig. 5). Detection of putative false linkages in branching points can be done based on the following: If the density of markers is high then linkages are usually proven by linkages of neighbor markers. This means that if some linkages are unproven by linkages of neighbor markers then such linkages can be false. Indeed, real deviations from linear structure are biologically possible (e.g., chromosomes of bacteria have a circular structure; chromosomal distortion can result in branching that highly reduces the fitness and fertility) but are quite rare.

Additional filtration of markers can be done after building of linkage groups and draft ordering. In practical, if local ordering of markers is not robust to jackknife resampling of data then sum markers in these regions have presumably higher proportion of errors in genotypes and should be excluded.

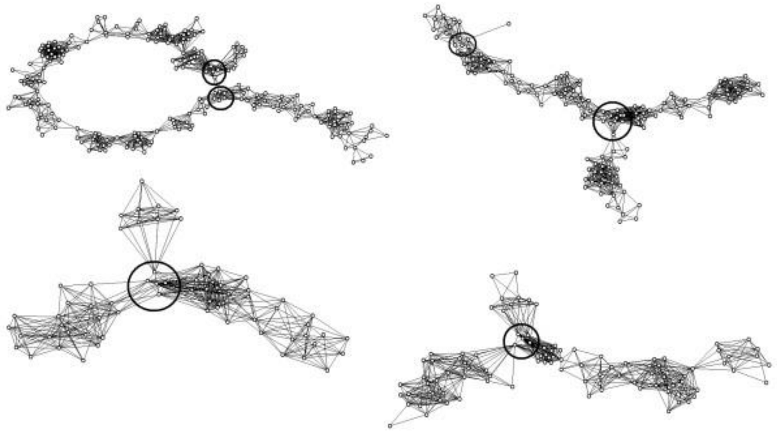


Figure 5. False linkages causing deviation from the linear structure of the chromosome. Here nodes are corresponding to markers. Edges are corresponding to linkages at some cutoff. Problematic regions (with branching) are indicated by circles. Usually such branchings are caused by single markers. Excluding of such markers leads to splitting of linkage groups onto linear parts [22].

2.7.2 Grouping markers onto groups of closely related markers

To reduce the number of markers, we can select the best delegates out of groups of closely related markers. For this purpose we use k-means clustering with initial centroids selected based on distant markers. After k-means clustering iterations, from each cluster we take only markers corresponding to centroids. For such markers some genotypes can be imputed or even corrected based on other markers from the cluster.

2.7.3 Estimation of recombination rates

We characterize genetic linkage between two markers m_1 and m_2 by recombination rate $r = r_{m1m2}$. Observed recombination rate is scored by proportion of individuals with an odd number of recombination between m_1 and m_2 among all individuals with known genotypes in both m_1 and m_2: $r = n_r /(n_r+n_n)$. Here n_r is the observed number of recombinants and n_n is the observed number of non-recombinants. In addition to such maximal likelihood estimation (MLE) we also need to estimate the confidence interval. Limits of this interval can be calculated as roots of equations: $\Pr(N_r>r_{\max}\ n)=\alpha/2$ and $\Pr(N_r>r_{\min}\ n)=\alpha/2$.

2.8 Cluster analysis

Clustering is a Machine Learning technique that involves the grouping of data points. Given a set of data points, we can use a clustering algorithm to classify each data point into a specific group. Data points that are in the same group should have similar properties and/or features, while data points in different groups should have highly dissimilar properties and/or features. Clustering is a method of unsupervised learning (consisting of input data without labeled responses) and is used in many fields. Various algorithms, proposed for genetic mapping datasets, employ the ideas of clustering to unite closely situated markers on the chromosome (to reduce the problem size and to improve the data quality) and to subdivide markers onto linkage groups (that are expected to be corresponding to chromosomes).

2.8.1 Single-linkage clustering

Single-linkage clustering is one of several methods of hierarchical clustering that measures the minimum distance between two clusters. Clustering using single linkage tends to produce an effect called chaining where single genes are added to clusters one at a time which is the opposite of complete linkage that measures the distance between the farthest two points in the clusters. Subdivision of markers onto linkage groups by single linkage with some cutoff (e.g., 0.15) warrants that tightly linked markers will be in the same linkage group while unlinked are expected to be in different linkage groups. This enables subdivision of the problem of genetic mapping onto independent problems with smaller numbers of markers.

2.8.2 K-Means Clustering

The K-Means clustering algorithm is known to be efficient for a small number of clusters or in the case of some heuristics for initial centroids selection. This clustering algorithm was developed by McQueen, and is one of the simplest and the best known unsupervised learning algorithms that solve the well-known clustering problem. The k-means algorithm aims to partition a set of objects, based on their attributes/features, into k clusters, where *k* is a predefined or user-defined constant. The main idea is to define *k* centroids, one for each cluster. The centroid of a cluster is formed in such a way that it is closely related (in terms of similarity function; similarity can be measured by using different methods such as cosine similarity, Euclidean distance, Extended Jaccard, Compression distance) to all objects in that cluster. In genetic mapping k-means help to group markers of linkage groups into clusters of tightly linked markers. Usually it is impossible to discriminate between different orderings of markers tightly linked to a single genomic region. Hence, it is enough to select a single delegate from such a cluster (corresponding to the cluster's centroid). This centroid can be considered as a virtual marker. Genotype of this marker can be imputed and corrected based on other markers of the cluster.

2.9 Ordering of skeleton markers within linkage groups

Incorrect ordering of markers within a linkage group leads to an increase of the sum of recombination rates between consequent markers. Based on this idea, ordering of markers can be based on the principal that correct ordering is one minimizing this sum. Searching for such ordering is analogous to searching for the shortest path on the edges of the net that visits all nodes exactly once. It is a variant of the classical TSP problem.

2.9.1 Travelling salesman problem

The travelling salesman problem (also called TSP) asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?". It is an NP-hard problem in combinatorial optimization, important in operations research. TSP can be modelled as an undirected weighted graph, such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the edge's weight. It is a minimization problem starting and finishing at a specified vertex after having visited each other vertex exactly once (Fig. 6). Often, the model is a complete graph (i.e. each pair of vertices is connected by an edge). If no path exists between two cities, adding an arbitrarily long edge will complete the graph without affecting the optimal tour [23].

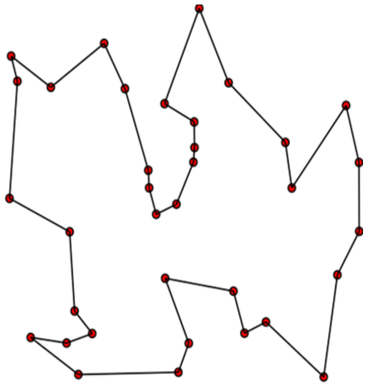


Figure 6. Example of cyclical path

Solution of practical TSP usually includes heuristics that are reasonable for the dealing data. In the case of genetic mapping it is reasonable to try to use local path optimizations. Such local optimisations can include testing of all possible ordering of relatively small (e.g., from 2 to 7) number of markers (Brute Force, Naive Approach [24]. Another popular heuristic in this field is the nearest neighbor method for building initial solutions (entire or for the sum subset of nodes). The key to this method is to always visit the nearest destination and then go back to the first node when all other nodes are visited. To connect this partial solution to the other parts we need to try to exclude one edge of the resulting cycle and add corresponding connecting edges. In good situations (what we expect in genetic mapping) high quality initial solutions including all nodes can be constructed based on a minimal spanning tree (MST) that can be found within time O(n^3). Further optimisation can be based on Branches and Bounds (B&B) methods or based on Genetic Algorithm (GA).

2.9.2 Minimum Spanning Tree

In graph theory, the minimal spanning tree (MST) is a connected subgraph including all nodes, without cycles and having the minimal possible sum of edges weights (Fig. 7) [25]. In the general case it is not unique. Real chromosomes have one-dimensional structure. Markers closely situated in the chromosome should be tightly linked while recombination rates between distant markers should be much higher. In such a situation, MST should have a topological structure including a long path along the chromosome and short branches to markers that are not included in this path. Initial solutions for TSP can be constructed from this long path and cyclical paths on markers from the branches. In more complicated cases (with longer branches) this approach can be applied recurrently but usually it is the result of false linkages that should be somehow filtered out.

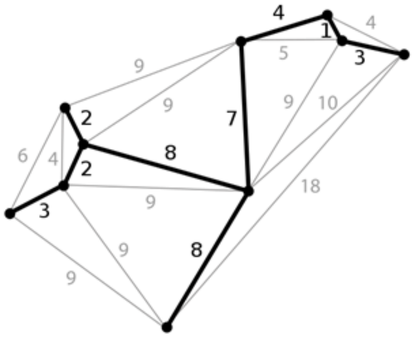
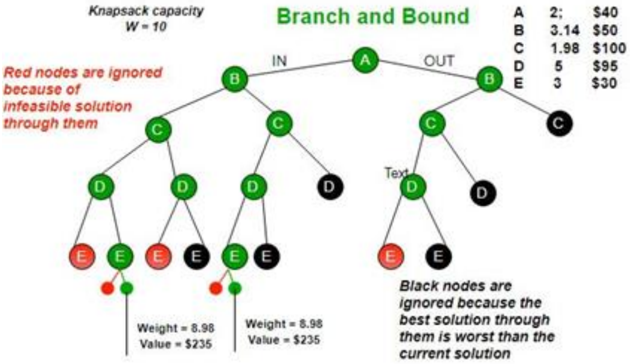


Figure 7. Minimal spanning tree

2.9.3 The Branch and Bounds

The Branches and Bounds (B&B) approach is based on the principle that the total set of feasible solutions can be partitioned into smaller subsets of solutions. These smaller subsets can then be evaluated systematically until the best solution is found. B&B is generally used for solving combinatorial optimization problems. These problems are typically exponential in terms of time complexity and may require exploring all possible permutations in the worst case. B&B technique solves these problems relatively quickly (Fig. 8).



Knapsack capacity
W = 10
Branch and Bound
A 2; $40
B 3.14 $50
C 1.98 $100
D 5 $95
E 3 $30
IN
OUT
A
B
B
C
C
C
C
Red nodes are ignored because of infeasible solution through them
D
D
D
D
Text
D
D
E
E
E
E
E
E
E
E
Weight = 8.98
Value = $235
Weight = 8.98
Value = $235
Black nodes are ignored because the best solution through them is worst than the current solution

Figure 8. Example of B&B approach application [26]

2.10 Genetic algorithm

Genetic Algorithms (GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems. Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to the next generation [27]. In simple words, they simulate “survival of the fittest” among individuals of consecutive generations for solving a problem. Each generation consists of a population of individuals and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits (in our case, individual is a variant of marker ordering). This string is analogous to the chromosome (see 2.2-2.4).

2.11 Short review of existing packages

MultiPoint is a tool developed by MultiQTL [28]. MultiPoint is an interactive system for building reliable genetic maps for thousands of markers. The main advantage of MultiPoint is that resulted solutions are robust to small changes in input data and resulted genetic maps have realistic lengths. The main disadvantage is that usage of this package calls for a lot of manual work of qualified users. Most biologists find it difficult to use such software on large data with errors.

The **Concorde TSP Solver** is a program for solving the travelling salesman problem. It was written by David Applegate, Robert E. Bixby, Vašek Chvátal, and William J. Cook, in ANSI C, and is freely available for academic use. This software provides a good solution for the intermediate (up to 3000) number of markers with about no errors [29]. Correlated errors and missing data make results of this program impractical for genetic mapping.

MSTmap is a software tool that is capable of constructing genetic linkage maps efficiently and accurately. It can handle various mapping populations including BC1, DH, Hap, and RIL, among others. The tool builds fastly the genetic linkage map by first constructing a Minimum Spanning Tree (MST), and hence the name MSTmap. The main disadvantage of this package is low robustness to errors and missing data. Missing data and errors result in false tight linkages. These false linkages result in global errors on genetic maps. In particular, solutions of this tool usually consist of incorrectly merged blocks of real linkage groups [30].

Geneious [31] . It enables building a good genetic map in the case of good, not too big data. In the case of substantial amounts of correlated errors this package meets unsolvable difficulties.

3. ALGORITHM

3.1 Data format:

3.1.1. Genotypes:

Genotypes usually are stored in the text/csv files within the tables having the following structure:

Marker	1	2	3	4	5 ...
M1	0	1	0	0	1
M2	1	1	0	0	1
M3	0	-	0	0	0

….

Here columns are individuals of the mapping population (*n*~50-300); rows are markers (N_m ~ from 20 to 10,000,000). In simple cases (what our project deals with) genotypes of markers have only two states (alleles 0 and 1); data on some genotypes can be missed (marked by “-”).

3.1.2. Genetic map

In addition to graphical representations, genetic maps are stored in tables (text/csv) generally with the following format:

ID	Marker Name	Linkage ID	Chromosome	Genetic coordinates
1	M1	1	chr1	20
2	M2	1	chr1	30.5
3	M3	1	chr1	55.7
4	M4	2	chr2	4

Here rows are corresponding to markers. Column 1 represents markers ID, Column 2 represents markers names (M1,M2,M3,M4, M5,...). Column 3,4 represents linkage group ID and chromosome for the marker (chr1,chr2,...). Column 4 represents genetic coordinates on the chromosome. These coordinates are scored in centiMorgans (cM) and can be used for calculation of expected recombination rates. For example, genetic distance between markers M1 and M2 from chromosome chr1 is d=|30.5-20|. Following the Haldane formula,
r = 0.5(1-exp(-0.02 d)) (see 2.4.1).

3.2 Input and output of our program

Input of our program is the file with genotypes. We also can upload genetic maps constructed by other tools or with using different options and parameters in the analysis. Output of our program will consist of the following:

- Tables with statistics on markers (missing data, allele frequencies,...)
- Graphs with distributions
- Network (graph) of linkages in format of Pajek
- Table and linart for comparison of map-based v.s. observed recombination rates
- List of markers that need to exclude to obtain linear clusters (based on recombination rates for neighbour markers)
- Table of linkage groups
- Table of skeleton markers
- Table with positions for all markers

3.3 Algorithm for building genetic map

```
0.  Input genotypes, check data format
1.  Calculate statistics on markers: proportion of missing data $p_{\mathrm{miss}}=n_{\mathrm{missed}}/n_{\mathrm{inds}}$, allele frequencies $f=n_1/(n_1+n_0)$, allele segregation quality $X^2=2(n_0+n_1)(f-0.5)^2/0.5$
2.  Filter out low quality markers ($p_{\mathrm{miss}}>20\%$ or $X^2>3.84$)
3.  Compare markers: identification of groups of tightly linked markers ($\max(n_{00},n_{01})\leq2$)
4.  Select the best delegate from each group: lowest average $\max(n_{00},n_{01})$, lowest $X^2$, lowest $p_{\mathrm{miss}}$.
5.  Impute/improve genotypes based on other markers from tightly linked group: minimization of average $\max(n_{00},n_{01})$
6.  Calculate recombination rates: $r=\min(n_{01}+n_{10},n_{00}+n_{11})/(n_{00}+n_{01}+n_{10}+n_{11})$
7.  Build graph (network) of linkages: nodes correspond to markers, edges correspond to linkages with $r<r_0=0.2$.
8.  Subdivide the network onto linked parts $LG_i$, If linked parts are too large (>40% of net), try reduce $r_0$ on 0.01 (Go to 7).
9.  For selected $LG_i$ :
    9.1  Save network in Pajek format
    9.2  Find Minimal Spanning Tree (MST)
    9.3  Find the longest path of MST
    9.4  Calculate ranks of nodes relatively to the longest path (based on the network)
    9.5  If all ranks not exceed 2 then this linkage group is already good and the longest path is the ordered list of skeleton markers (stop). Else:
         Option 1: Inspect network structure using Pajek software, manually exclude nodes and edges causing branches and cycles. Go to 8.
         Option 2: Check linkages of linked markers. Exclude nodes and edges unproven by parallel linkages (i.e., such that after excluding each of them, the network of linked nodes loses connectivity. Go to 8.
10. Try end-to-end merge $LG_i$ by increasing $r_0$ on 0.01 (Go to 7).
11. Output resulting linkage groups and paths in format of tables
12. Find the optimal position on the resulting map for all unmapped markers: for each interval of paths calculate recombination rates with markers on the ends, select the one with minimal sum, estimate position on the interval by one maximizing the likelihood.
13. Output the final map in table format.
```

4. GOALS

The main goal of our project is to improve the filtering and clustering stages of the MultiPoint algorithm for genetic mapping. In particular we (i) impute missing data for markers having tightly linked neighbours, (ii) filter out linkages unproven by linkages of neighbour markers, (iii) filter out markers resulted in non-linear topological structure of the network of linkages, (iv) control topological structure of network of genetic linkages to provide recommendations to merge or split linkage groups (Pajek, http://mrvar.fdv.uni-lj.si/pajek/), output genetic map for skeleton markers defined as one from the longest path of the corresponding MST.

The second goal is provide quick and efficient data processing and network build time even in case of huge number of markers

The third goal is to check efficiency of our improvements for genetic map constructions for real data.

5. PRELIMINARY SOFTWARE ENGINEERING DOCUMENTATION

5.1. Requirements (Use-Case)

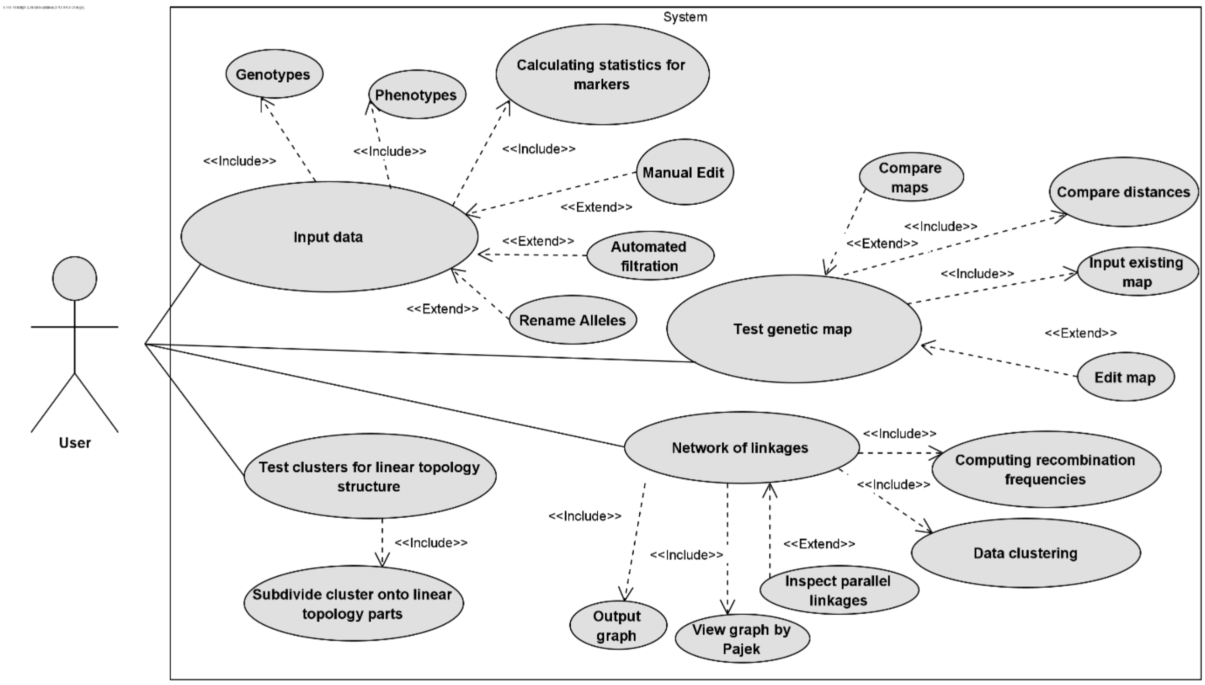


Figure 9. Use case diagram

UC1: Input Data

- Goal: Input data of genotypes/phenotypes for program analysis and processing.
- Preconditions: Valid path with .txt input file
- Possible user errors: Input file of invalid format
- Limitations: Browsed files must be only ".txt" format, that contain valid genetic map data
- Pseudo code Flow:

Actor	System
1) Browse file from file explorer	2) Highlights the selected file
3) Click "Ok" to confirm file path	4) Save path of selected file and load for analysis

UC2: Test genetic map

- Goal: Compare distances/edit map/calculate criterion
- Preconditions: Input data already loaded from UC1
- Possible user errors: Invalid map data.
- Limitations: Valid map format
- Pseudo code Flow:

Actor	System
1) Click "Import genetic map"	2) Browser for genetic map path
3) Click "Ok" after choosing path	4) Load genetic map

UC3: Network Of Linkages

- Goal: Output a graph of linkages in Pajek format to a specific path
- Preconditions: Loaded and valid input data from UC1 and selected linkages for graphing
- Possible user errors: None.
- Limitations: None
- Pseudo code Flow:

Actor	System
1) Click "Export to a Pajek graph"	2) File browser for save path
3) Click "Ok" to confirm output path	4) Create a Pajek format graph at save path

UC4: Test clusters for linear topology structure

- Goal: Test clusters for linear topology structure based on ranks relative to the longest path of MST and subdivide if needed
- Preconditions: Loaded and valid input data from UC1
- Possible user errors: None.
- Limitations: None
- Pseudo code Flow:

Actor	System
1) Click "Test topology structure"	2) Calculate and show results in graph

5.2 Class diagram

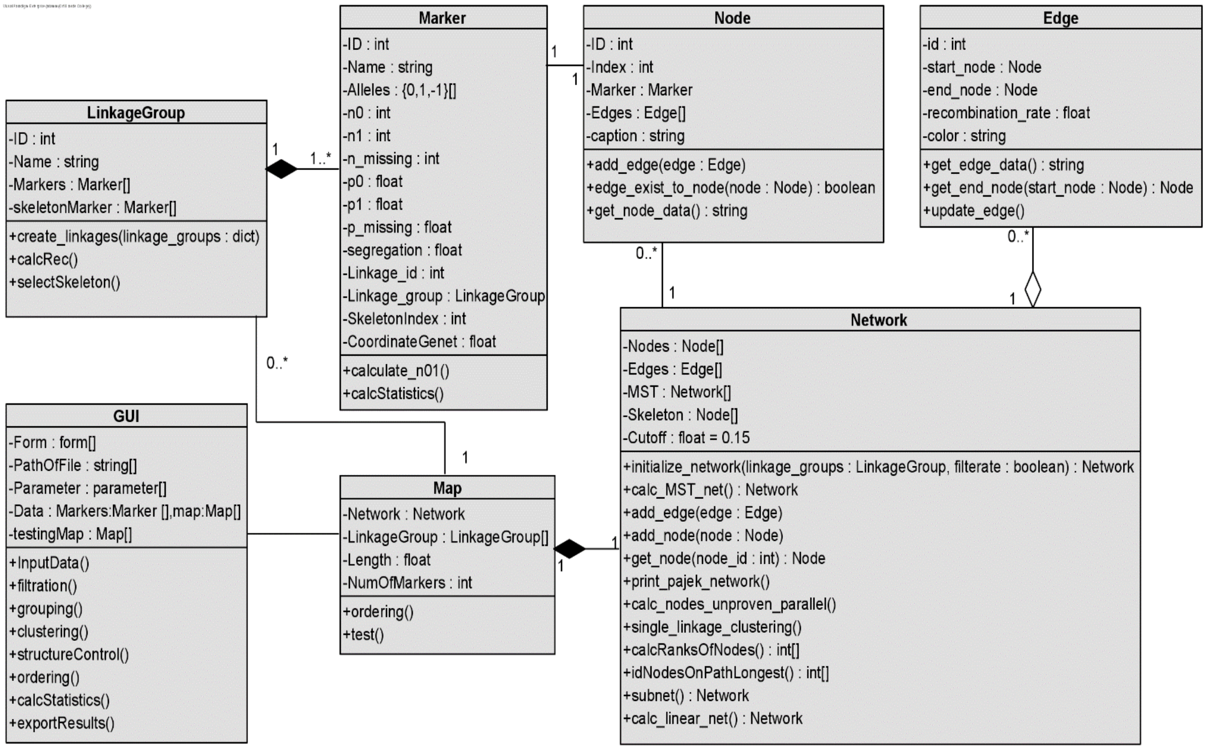
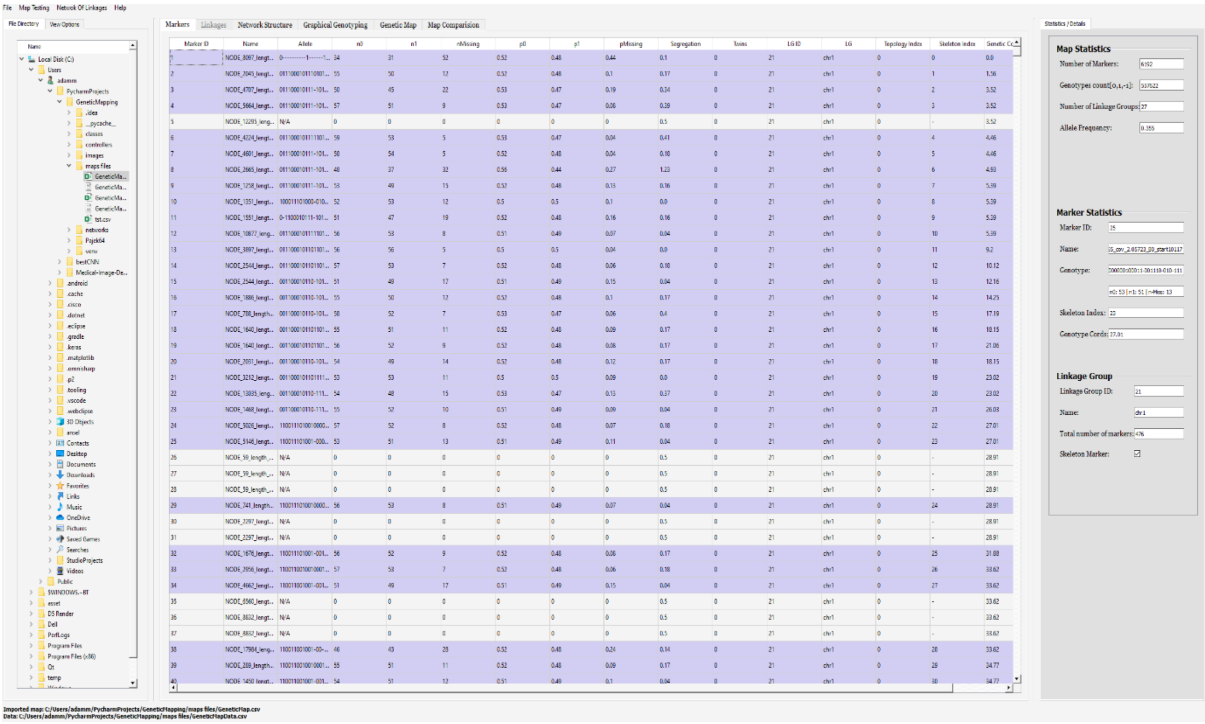


Figure 10. Class diagram

5.3 GUI Design

Our GUI design is based on 3 split panes: Left side is a File Browser and more view options, Middle is our main display screen where lies our map editor and main functionality, on the right side we display statistics regarding the displayed data (Fig. 11). **Markers tab** contains information on markers, their names, alleles, linkage group and more information.(Fig. 11). In the **Linkages tab** we show statistics of a selected marker compared to its linkage with other markers of its linkage group(Fig. 12). **Network Structure tab** represents our main algorithm and visualization of network structure for the selected linkage group(Fig. 14). These graphs are processed in the software and the result is presented as (2D/3D) plot in Pajek software (Fig. 15 and 16). **Graphical Genotyping tab** enables graphical control of the resulted marker ordering within the linkage group and allows renaming alleles for data correction (Fig. 18 and 19). **Genetic Map tab** enables users to see the imported genetic map (3.1.1), in the format of genotypes (Fig. 13). Genetic map editor helps to modify the genetic map data and display data based on modified map. **File Browser** enables users to navigate through the system to load files into the software to process. **Statistics** displays general statistics of genetic map/markers/linkage groups. (Fig. 20).



Marker ID	Name	Allele	n0	n1	nMissing	p0	p1	pMissing	Segregation	Twins	LG ID	LG	Topology Index	Skeleton Index	Genetic Cc
1	NODE_8097_lengt...	0----------1------1...	34	31	52	0.52	0.48	0.44	0.1	0	21	chr1	0	0	0.0
2	NODE_2045_lengt...	0111000101110101...	55	50	12	0.52	0.48	0.1	0.17	0	21	chr1	0	1	1.56
3	NODE_4707_lengt...	011100010111-101...	50	45	22	0.53	0.47	0.19	0.34	0	21	chr1	0	2	3.52
4	NODE_5664_lengt...	011100010111-101...	57	51	9	0.53	0.47	0.08	0.39	0	21	chr1	0	3	3.52
5	NODE_12295_leng...	N/A	0	0	0	0	0	0	0.5	0	21	chr1	0	-	3.52
6	NODE_4224_lengt...	0111000101111101...	59	53	5	0.53	0.47	0.04	0.41	0	21	chr1	0	4	4.46
7	NODE_4601_lengt...	011100010111-101...	58	54	5	0.52	0.48	0.04	0.18	0	21	chr1	0	5	4.46
8	NODE_2665_lengt...	011100010111-101...	48	37	32	0.56	0.44	0.27	1.23	0	21	chr1	0	6	4.93
9	NODE_1258_lengt...	011100010111-101...	53	49	15	0.52	0.48	0.13	0.16	0	21	chr1	0	7	5.39
10	NODE_1351_lengt...	100011101000-010...	52	53	12	0.5	0.5	0.1	0.0	0	21	chr1	0	8	5.39
11	NODE_1551_lengt...	0-1100010111-101...	51	47	19	0.52	0.48	0.16	0.16	0	21	chr1	0	9	5.39
12	NODE_10877_leng...	0111000101111101...	56	53	8	0.51	0.49	0.07	0.04	0	21	chr1	0	10	5.39
13	NODE_3897_lengt...	0111000101101101...	56	56	5	0.5	0.5	0.04	0.0	0	21	chr1	0	11	9.2
14	NODE_2544_lengt...	0111000101101101...	57	53	7	0.52	0.48	0.06	0.18	0	21	chr1	0	12	10.12
15	NODE_2544_lengt...	001100010110-101...	51	49	17	0.51	0.49	0.15	0.04	0	21	chr1	0	13	12.16
16	NODE_1886_lengt...	001100010110-101...	55	50	12	0.52	0.48	0.1	0.17	0	21	chr1	0	14	14.25
17	NODE_788_length...	001100010110-101...	58	52	7	0.53	0.47	0.06	0.4	0	21	chr1	0	15	17.19
18	NODE_1640_lengt...	0011000101101101...	55	51	11	0.52	0.48	0.09	0.17	0	21	chr1	0	16	18.15
19	NODE_1640_lengt...	0011000101101101...	56	52	9	0.52	0.48	0.08	0.17	0	21	chr1	0	17	21.06
20	NODE_2031_lengt...	001100010110-101...	54	49	14	0.52	0.48	0.12	0.17	0	21	chr1	0	18	18.15
21	NODE_3212_lengt...	0011000101101111...	53	53	11	0.5	0.5	0.09	0.0	0	21	chr1	0	19	23.02
22	NODE_13035_leng...	001100010110-111...	54	48	15	0.53	0.47	0.13	0.37	0	21	chr1	0	20	23.02
23	NODE_1468_lengt...	001100010110-111...	55	52	10	0.51	0.49	0.09	0.04	0	21	chr1	0	21	26.03
24	NODE_5026_lengt...	1100111010010000...	57	52	8	0.52	0.48	0.07	0.18	0	21	chr1	0	22	27.01
25	NODE_5146_lengt...	110011101001-000...	53	51	13	0.51	0.49	0.11	0.04	0	21	chr1	0	23	27.01
26	NODE_59_length_...	N/A	0	0	0	0	0	0	0.5	0	21	chr1	0	-	28.91
27	NODE_59_length_...	N/A	0	0	0	0	0	0	0.5	0	21	chr1	0	-	28.91
28	NODE_59_length_...	N/A	0	0	0	0	0	0	0.5	0	21	chr1	0	-	28.91
29	NODE_741_length...	1100111010010000...	56	53	8	0.51	0.49	0.07	0.04	0	21	chr1	0	24	28.91
30	NODE_2297_lengt...	N/A	0	0	0	0	0	0	0.5	0	21	chr1	0	-	28.91
31	NODE_2297_lengt...	N/A	0	0	0	0	0	0	0.5	0	21	chr1	0	-	28.91
32	NODE_1676_lengt...	110011101001-001...	56	52	9	0.52	0.48	0.08	0.17	0	21	chr1	0	25	31.88
33	NODE_2956_lengt...	1100110010010001...	57	53	7	0.52	0.48	0.06	0.18	0	21	chr1	0	26	33.62
34	NODE_4662_lengt...	110011001001-001...	51	49	17	0.51	0.49	0.15	0.04	0	21	chr1	0	27	33.62
35	NODE_6560_lengt...	N/A	0	0	0	0	0	0	0.5	0	21	chr1	0	-	33.62
36	NODE_8832_lengt...	N/A	0	0	0	0	0	0	0.5	0	21	chr1	0	-	33.62
37	NODE_8832_lengt...	N/A	0	0	0	0	0	0	0.5	0	21	chr1	0	-	33.62
38	NODE_17984_leng...	110011001001-00-...	46	43	28	0.52	0.48	0.24	0.14	0	21	chr1	0	28	33.62
39	NODE_289_length...	1100110010010001...	55	51	11	0.52	0.48	0.09	0.17	0	21	chr1	0	29	34.77
40	NODE_1450_lengt...	110011001001-001...	54	51	12	0.51	0.49	0.1	0.04	0	21	chr1	0	30	34.77

Map Statistics

Number of Markers: 6192

Genotypes count[0,1,-1]: 557522

Number of Linkage Groups: 27

Allele Frequency: 0.355

Marker Statistics

Marker ID: 25

Name: 35_cov_2.05723_80_start10117

Genotype: 000000100011-001110-010-111

n0: 53 | n1: 51 | n-Miss: 13

Skeleton Index: 23

Genotype Cords: 27.01

Linkage Group

Linkage Group ID: 21

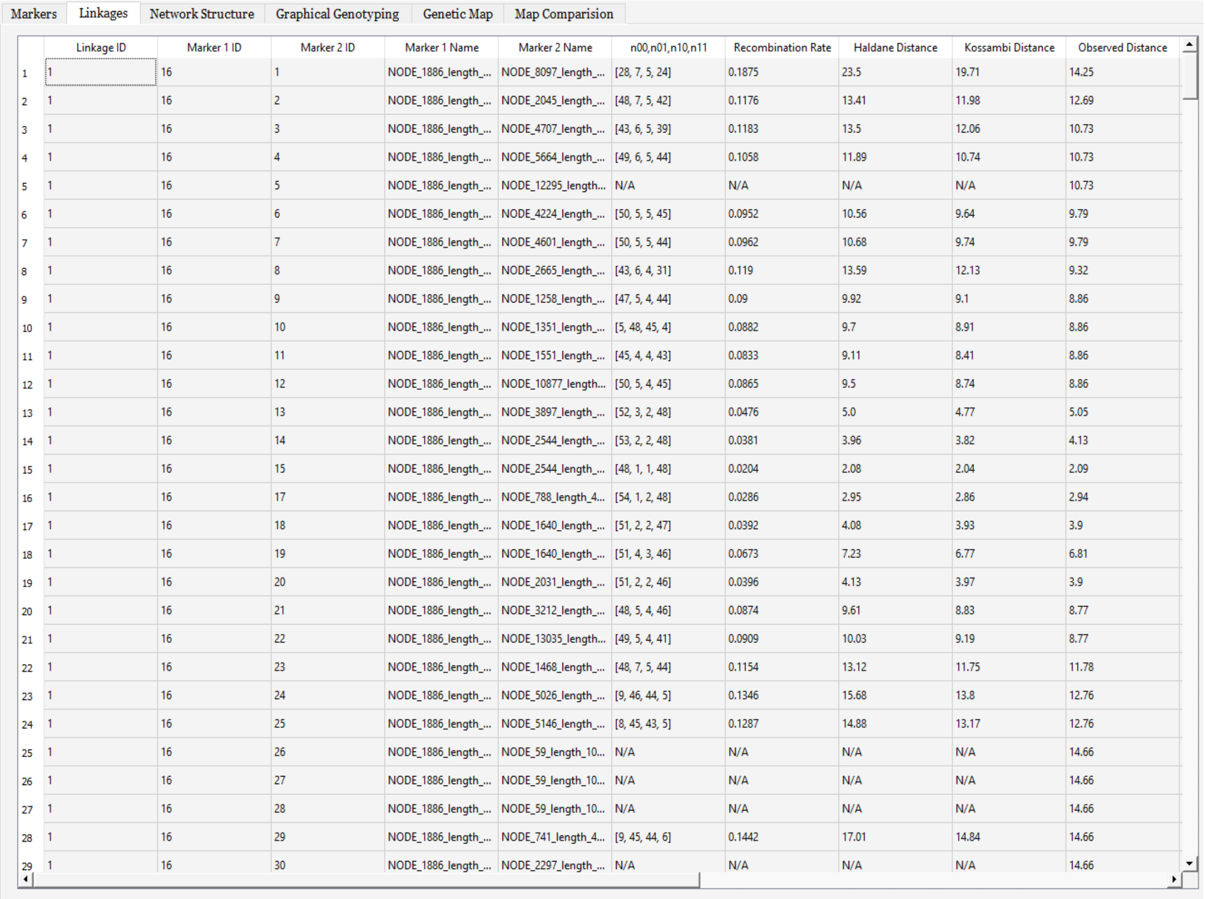
Name: chr1

Total number of markers: 476

Skeleton Marker: ☑

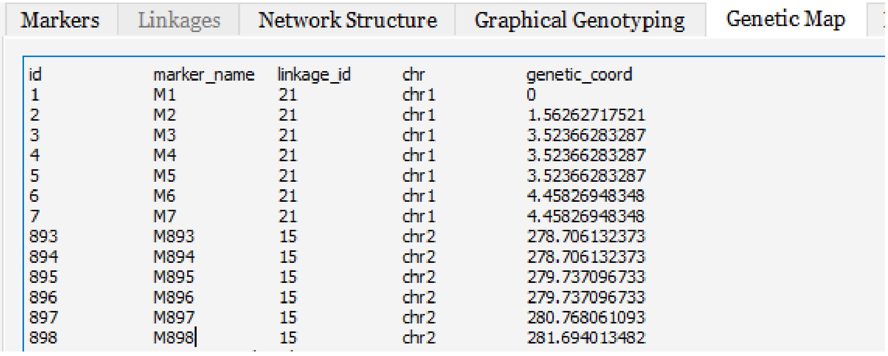
Imported map: C:/Users/adamm/PycharmProjects/GeneticMapping/maps files/GeneticMap.csv
Data: C:/Users/adamm/PycharmProjects/GeneticMapping/maps files/GeneticMapData.csv

Figure 11. The main interface showing file browser pane on the left, data and tabs navigation through functionalities in center pane and loaded map and selected marker statistics on right pane



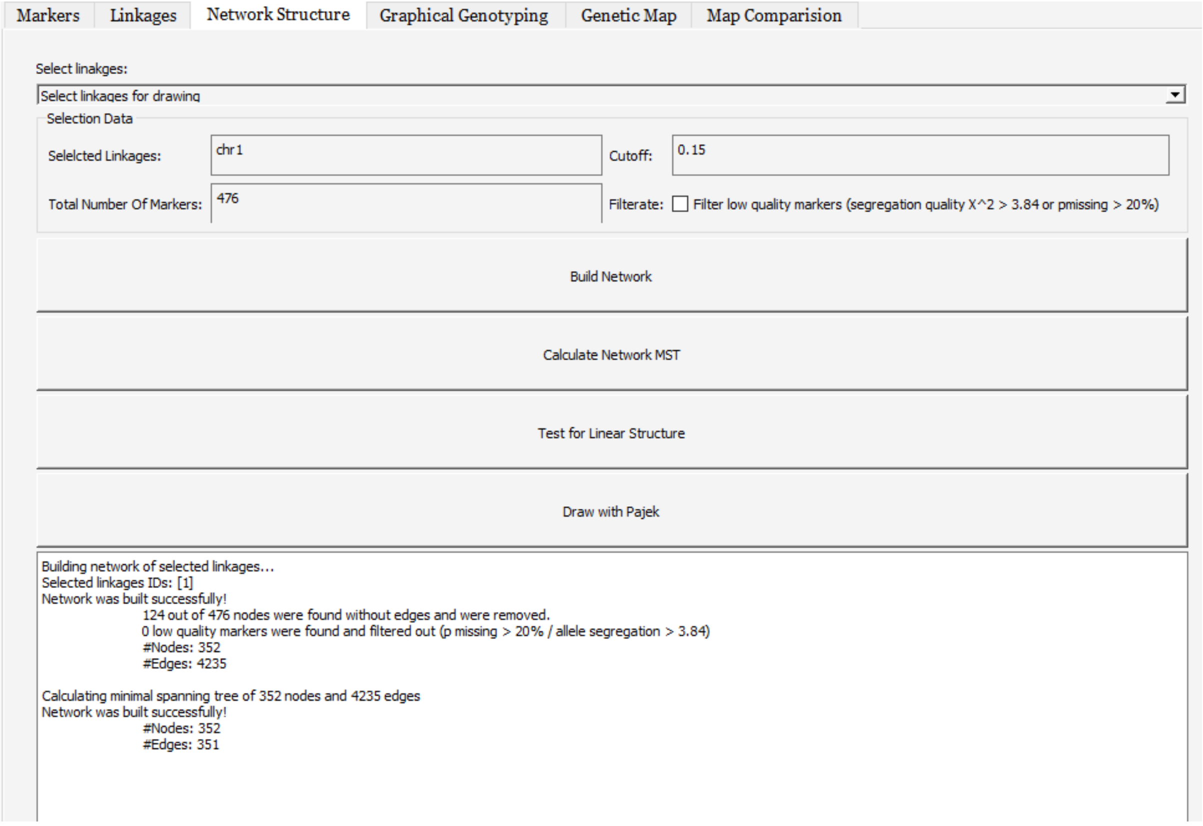
	Linkage ID	Marker 1 ID	Marker 2 ID	Marker 1 Name	Marker 2 Name	n00,n01,n10,n11	Recombination Rate	Haldane Distance	Kossambi Distance	Observed Distance
1	1	16	1	NODE_1886_length_...	NODE_8097_length_...	[28, 7, 5, 24]	0.1875	23.5	19.71	14.25
2	1	16	2	NODE_1886_length_...	NODE_2045_length_...	[48, 7, 5, 42]	0.1176	13.41	11.98	12.69
3	1	16	3	NODE_1886_length_...	NODE_4707_length_...	[43, 6, 5, 39]	0.1183	13.5	12.06	10.73
4	1	16	4	NODE_1886_length_...	NODE_5664_length_...	[49, 6, 5, 44]	0.1058	11.89	10.74	10.73
5	1	16	5	NODE_1886_length_...	NODE_12295_length...	N/A	N/A	N/A	N/A	10.73
6	1	16	6	NODE_1886_length_...	NODE_4224_length_...	[50, 5, 5, 45]	0.0952	10.56	9.64	9.79
7	1	16	7	NODE_1886_length_...	NODE_4601_length_...	[50, 5, 5, 44]	0.0962	10.68	9.74	9.79
8	1	16	8	NODE_1886_length_...	NODE_2665_length_...	[43, 6, 4, 31]	0.119	13.59	12.13	9.32
9	1	16	9	NODE_1886_length_...	NODE_1258_length_...	[47, 5, 4, 44]	0.09	9.92	9.1	8.86
10	1	16	10	NODE_1886_length_...	NODE_1351_length_...	[5, 48, 45, 4]	0.0882	9.7	8.91	8.86
11	1	16	11	NODE_1886_length_...	NODE_1551_length_...	[45, 4, 4, 43]	0.0833	9.11	8.41	8.86
12	1	16	12	NODE_1886_length_...	NODE_10877_length...	[50, 5, 4, 45]	0.0865	9.5	8.74	8.86
13	1	16	13	NODE_1886_length_...	NODE_3897_length_...	[52, 3, 2, 48]	0.0476	5.0	4.77	5.05
14	1	16	14	NODE_1886_length_...	NODE_2544_length_...	[53, 2, 2, 48]	0.0381	3.96	3.82	4.13
15	1	16	15	NODE_1886_length_...	NODE_2544_length_...	[48, 1, 1, 48]	0.0204	2.08	2.04	2.09
16	1	16	17	NODE_1886_length_...	NODE_788_length_4...	[54, 1, 2, 48]	0.0286	2.95	2.86	2.94
17	1	16	18	NODE_1886_length_...	NODE_1640_length_...	[51, 2, 2, 47]	0.0392	4.08	3.93	3.9
18	1	16	19	NODE_1886_length_...	NODE_1640_length_...	[51, 4, 3, 46]	0.0673	7.23	6.77	6.81
19	1	16	20	NODE_1886_length_...	NODE_2031_length_...	[51, 2, 2, 46]	0.0396	4.13	3.97	3.9
20	1	16	21	NODE_1886_length_...	NODE_3212_length_...	[48, 5, 4, 46]	0.0874	9.61	8.83	8.77
21	1	16	22	NODE_1886_length_...	NODE_13035_length...	[49, 5, 4, 41]	0.0909	10.03	9.19	8.77
22	1	16	23	NODE_1886_length_...	NODE_1468_length_...	[48, 7, 5, 44]	0.1154	13.12	11.75	11.78
23	1	16	24	NODE_1886_length_...	NODE_5026_length_...	[9, 46, 44, 5]	0.1346	15.68	13.8	12.76
24	1	16	25	NODE_1886_length_...	NODE_5146_length_...	[8, 45, 43, 5]	0.1287	14.88	13.17	12.76
25	1	16	26	NODE_1886_length_...	NODE_59_length_10...	N/A	N/A	N/A	N/A	14.66
26	1	16	27	NODE_1886_length_...	NODE_59_length_10...	N/A	N/A	N/A	N/A	14.66
27	1	16	28	NODE_1886_length_...	NODE_59_length_10...	N/A	N/A	N/A	N/A	14.66
28	1	16	29	NODE_1886_length_...	NODE_741_length_4...	[9, 45, 44, 6]	0.1442	17.01	14.84	14.66
29	1	16	30	NODE_1886_length_...	NODE_2297_length_...	N/A	N/A	N/A	N/A	14.66

Figure 12. Linkages tab shows statistics of a selected marker compared to its linkage with other markers of its linkage group



id	marker_name	linkage_id	chr	genetic_coord
1	M1	21	chr1	0
2	M2	21	chr1	1.56262717521
3	M3	21	chr1	3.52366283287
4	M4	21	chr1	3.52366283287
5	M5	21	chr1	3.52366283287
6	M6	21	chr1	4.45826948348
7	M7	21	chr1	4.45826948348
893	M893	15	chr2	278.706132373
894	M894	15	chr2	278.706132373
895	M895	15	chr2	279.737096733
896	M896	15	chr2	279.737096733
897	M897	15	chr2	280.768061093
898	M898	15	chr2	281.694013482

Figure 13. Table of markers. Optionally, it can be shown in tabs Markers, Graphical genotyping and Genetic Map



Select linakges:

Select linkages for drawing

Selection Data

Selelcted Linkages: chr1

Cutoff: 0.15

Total Number Of Markers: 476

Filterate: □ Filter low quality markers (segregation quality X^2 > 3.84 or pmissing > 20%)

```
Building network of selected linkages...
Selected linkages IDs: [1]
Network was built successfully!
                124 out of 476 nodes were found without edges and were removed.
                0 low quality markers were found and filtered out (p missing > 20% / allele segregation > 3.84)
                #Nodes: 352
                #Edges: 4235

Calculating minimal spanning tree of 352 nodes and 4235 edges
Network was built successfully!
                #Nodes: 352
                #Edges: 351
```

Figure 14: The tab enables choosing linkage groups of imported data of markers in build a network based on it / calculate the minimal spanning tree of linkage and test for linear structure, multiple options are available such as determining a cutoff value to build the linkages based on and the ability filter out low quality markers(segregation quality $X^2 > 3.84\ or\ P_{miss} > 20\%$)

Visualizations of linkage groups are presented with Pajek software.

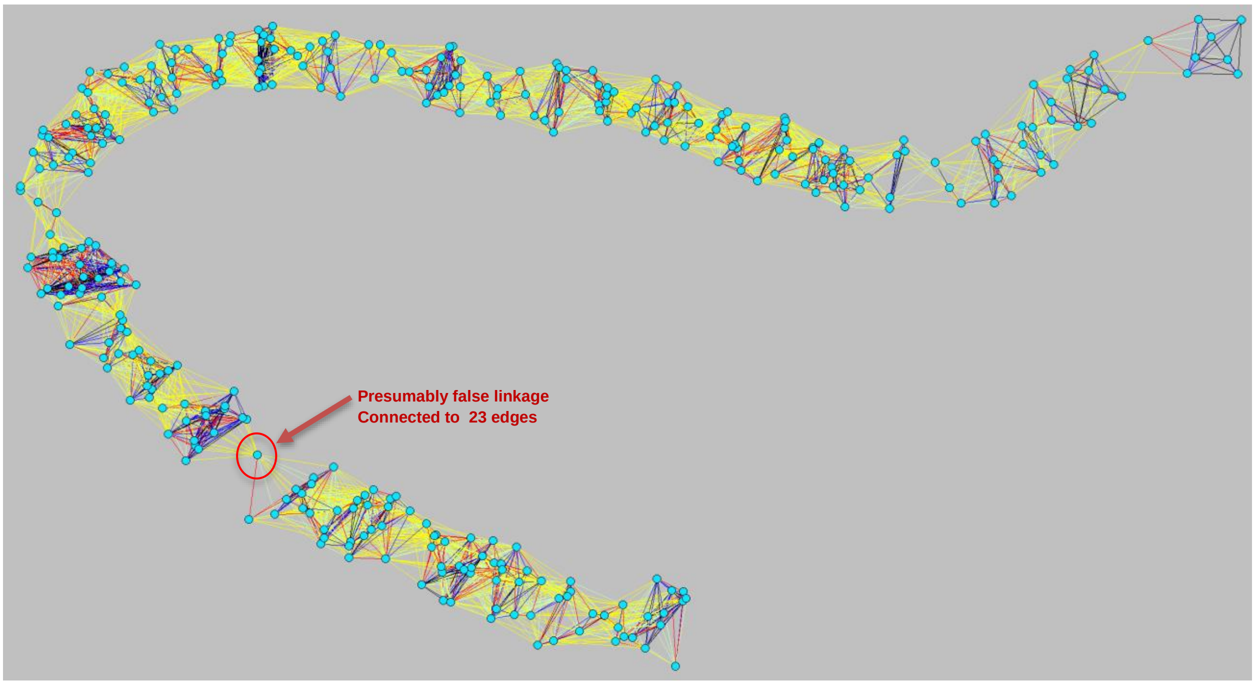


Figure 15. Pajek visualization of the selected linkage groups, here nodes correspond to markers. Edges correspond to linkages at some cutoff (0.15) and colors are based on recombination rate between markers. Highlighted node marks a deviation from the linear structure of the chromosome (like in Fig. 5) point on presumably false linkages or problematic markers (connected to 23 edges) which is aimed to be removed on subdivision to linear topology parts stage. A total of 352 nodes and 4235 edges.

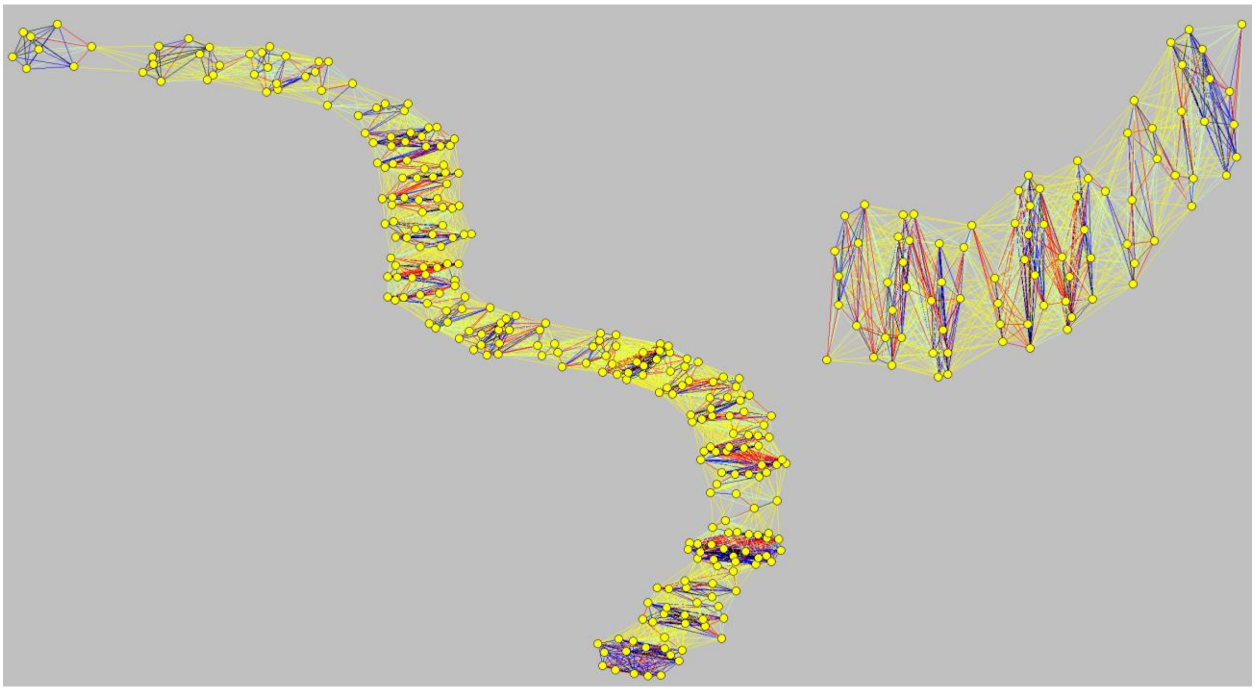


Figure 16. Pajek visualization of subdivided cluster from pervious figure where the false linkage (1 node and its 23 edges) is removed and the linkage group is divided into 2 linear topology structured parts based on parallel cutoff of 0.3 . A total of 351 nodes and 4212 edges.

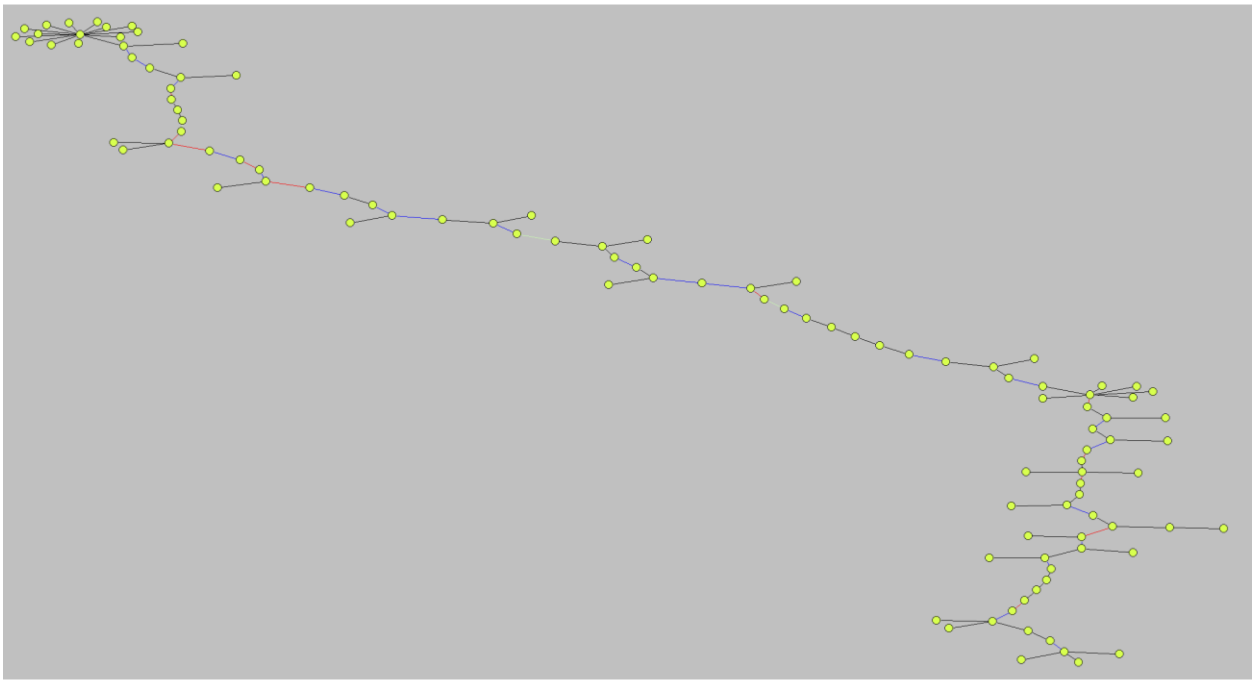


Figure 17. Minimal spanning tree of a linkage group that originally contains 109 nodes and 2237 edges. The output MST contains 109 nodes and 108 edges, MST is used to find skeletal shape of chromosome based on longest path.

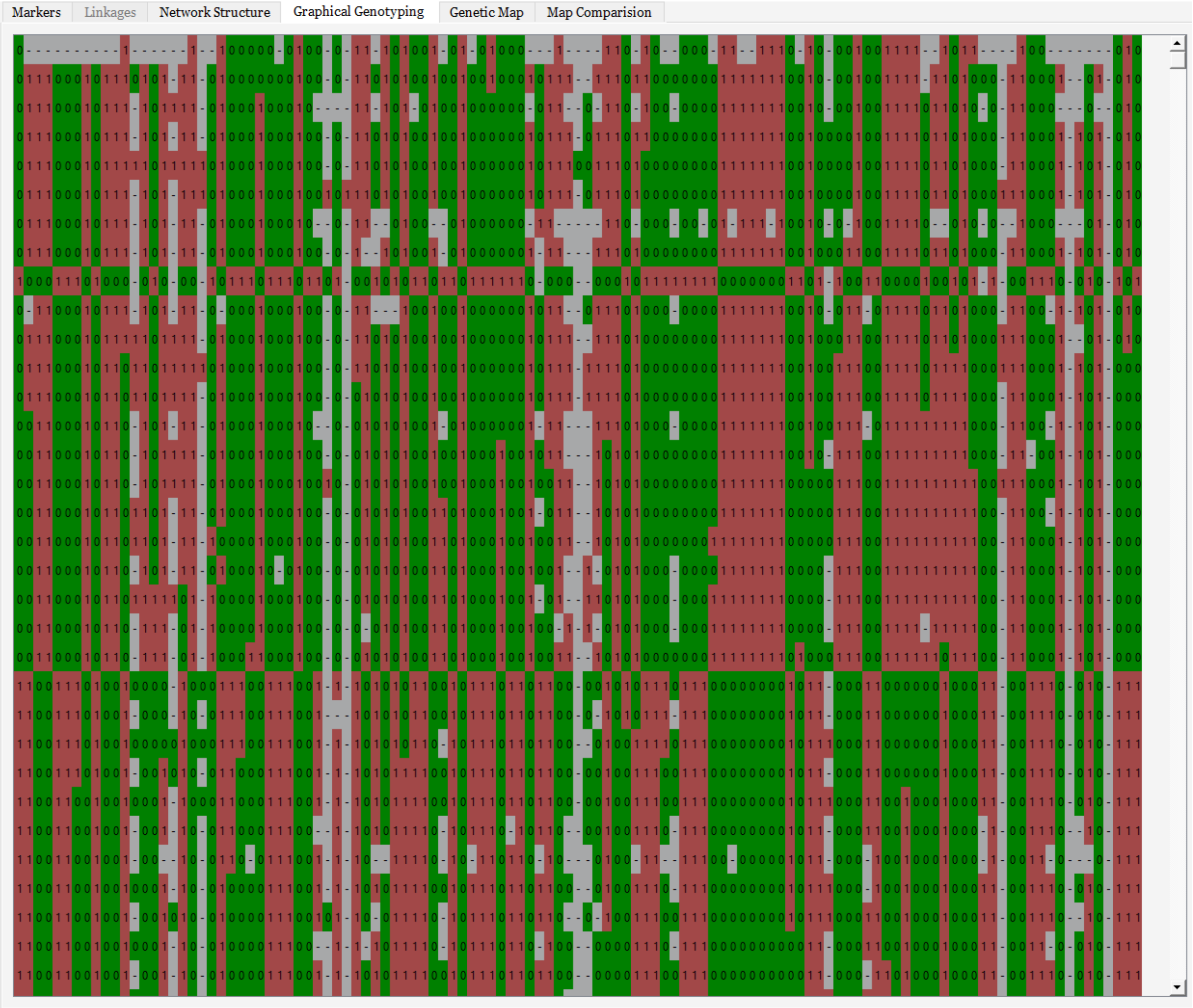


Figure 18. Graphical genotyping enables the characterization of the quality of the resulting genetic map (ordering of markers within linkage groups) graphically. Here colors represent genotypes within marker gray for missing data, green for allele 0, red for allele 1. In the case of backcross we expect allele segregation 0.5:0.5. In good situations we expect to have here relatively long vertical lines of the same color. Frequent changes between gray and red green colors points to putative error in ordering or too high proportion of errors in genotyping.

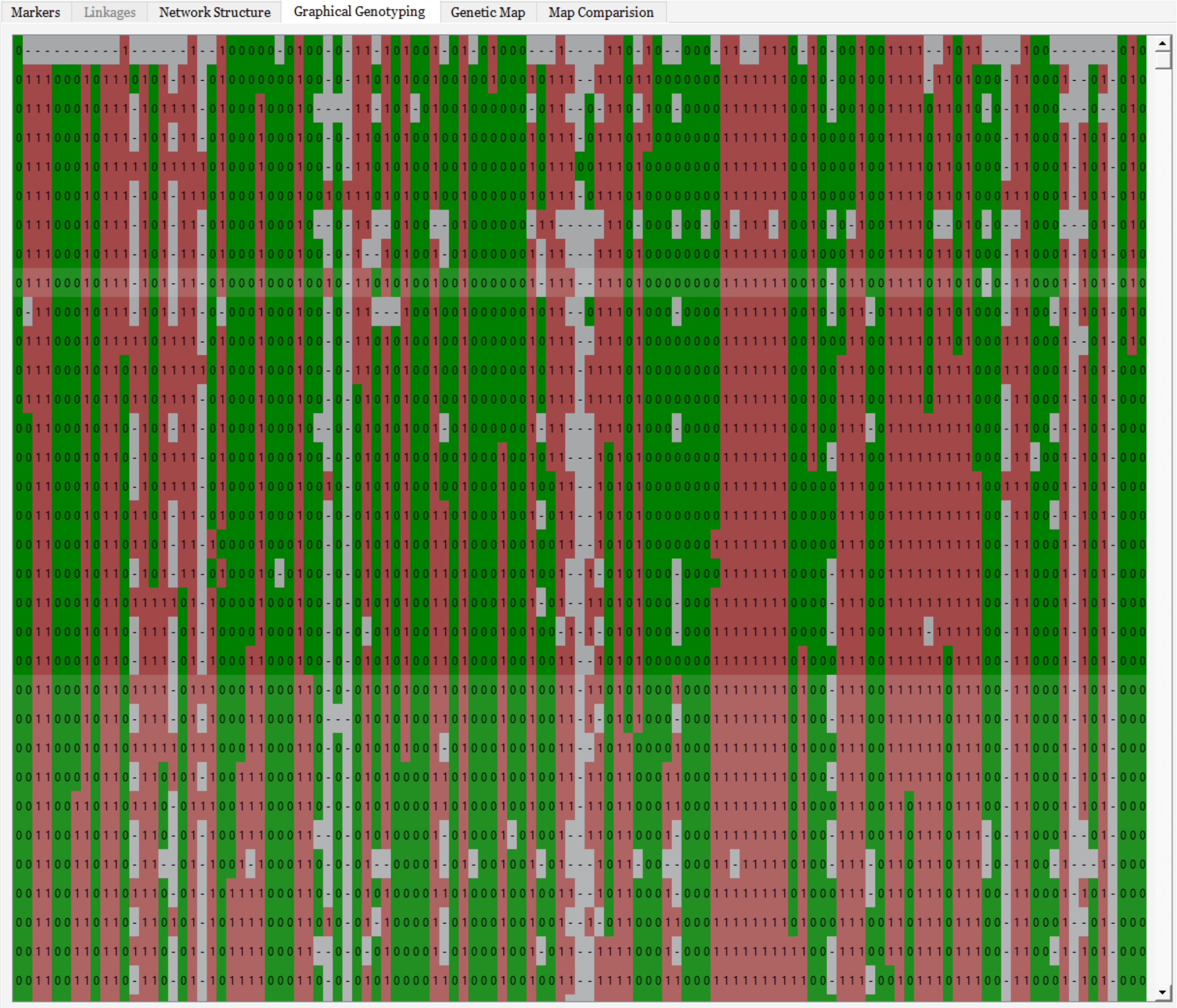
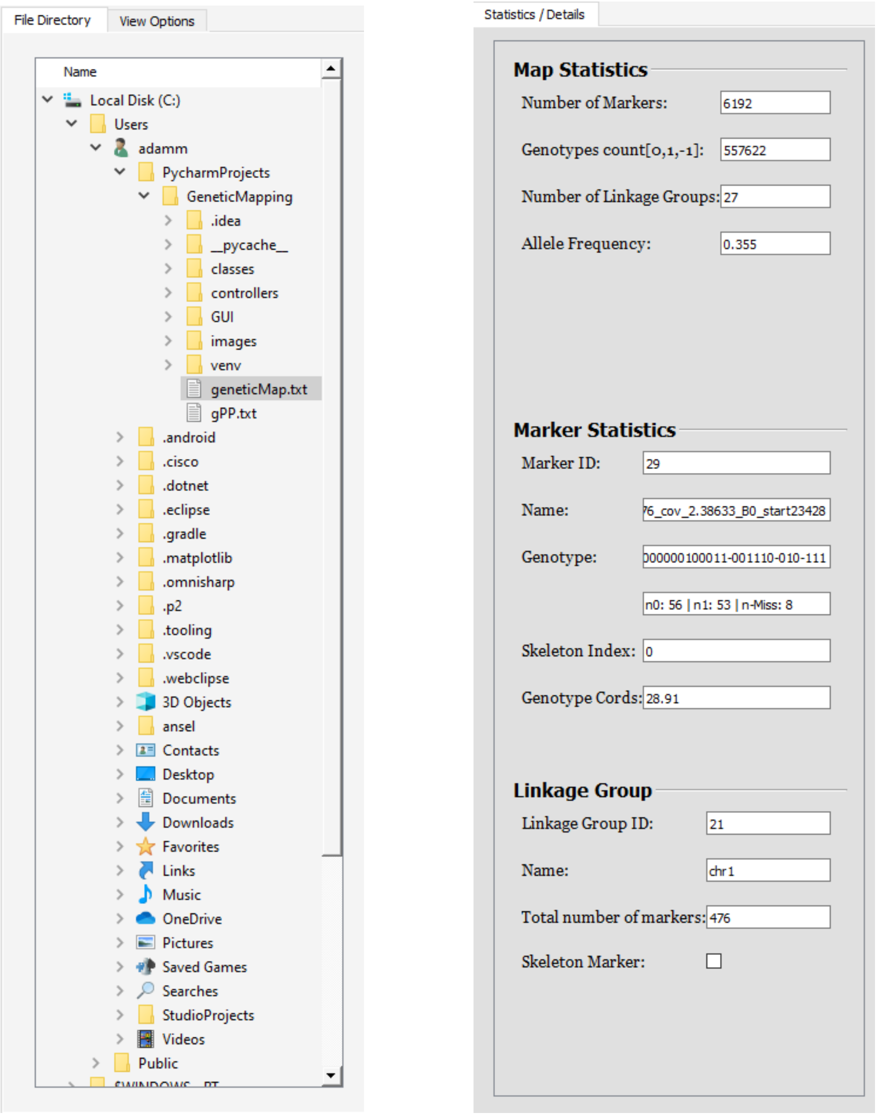


Figure 19. Genotypes after performing swaps for presumably false data providing a much more long vertical lines of same color, improving genotypes quality to presumably as expected. Swaps are based if the number of differences is bigger than the minimum number of 0s or number of 1s



Map Statistics

Number of Markers: 6192

Genotypes count[0,1,-1]: 557622

Number of Linkage Groups: 27

Allele Frequency: 0.355

Marker Statistics

Marker ID: 29

Name: 76_cov_2.38633_B0_start23428

Genotype: 000000100011-001110-010-111

n0: 56 | n1: 53 | n-Miss: 8

Skeleton Index: 0

Genotype Cords: 28.91

Linkage Group

Linkage Group ID: 21

Name: chr1

Total number of markers: 476

Skeleton Marker: □

Figure 20. File Browser (left) and Statistics (right) panes of GUI

5.4 Testing

Test #	Test Subject	Expected Result	Actual Results
	GUI Testing		
1.	Load Data and Display Statistics	Correct data statistics will be displayed on the right side of the main windows based on input	✓
2.	Navigation Between Results Windows	When changing between results windows, the window should display different kind of results	✓
3.	Export/Import button	Imports/Exports selected data in correct formatting	✓
	Algorithm Testing		
4.	Validate the newly created genetic map with an already created genetic map either manually or from another trusted program.	The resulting map would be similar to already existing one	✓
5.	Validate that the differences between importing an already existing genetic map and the calculated one form the algorithm.	The results of the already existing map will be similar to the one calculated with algorithm	✓
6.	Validate the filtered data from the algorithm and compare with a small manually filtered dataset.	Correctly filtered dataset	✓
7.	Validate that subdividing the network onto linked parts is correct with manual subdivide of the same graph	The network is subdivided the way it should compared to the manually subdivided graph	✓
8.	Validate that the recombination frequencies is calculated correctly with manually calculated frequencies or already known recombination frequencies	Correctly calculated frequencies	✓
9.	Validate the calculation of ranks of nodes relatively to the longest path (based on the network) with manual calculation of the same ranks	The ranks has been calculated correctly	✓
10.	Validate MST graph of output with a manually calculated MST graph performed on a small dataset	Identical MST trees	✓

6. RESULTS AND CONCLUSIONS

Results:

We tested our program on genetic data from a population of 117 ants from lab of Eyal Privman from University of Haifa.

In this data we have 4768 markers with their genotypes, we had too much missing data and had markers with highly non-symmetric segregation, after filtration we remained with 3565 markers. For these markers we calculated linkages based on these linkages we decided to use linkage cutoff 0.15 to subdivide markers onto linkage groups (27 linkage groups).

For each linkage group we've checked linearity of structure. To further improve the structure we used additional filtration for markers and linkages based on searching of parallel linkages (we searched with cutoff 0.20 for linkages lower 0.05) , based on this test we excluded 12 markers and 121 edges linkages. After search filtration we clustered again and obtained 30 linkage groups based on cutoff 0.15. These linkage groups were found having linear structure based on our network criteria. We also tested linear structure manually based on visualization of network structure using Pajek software.

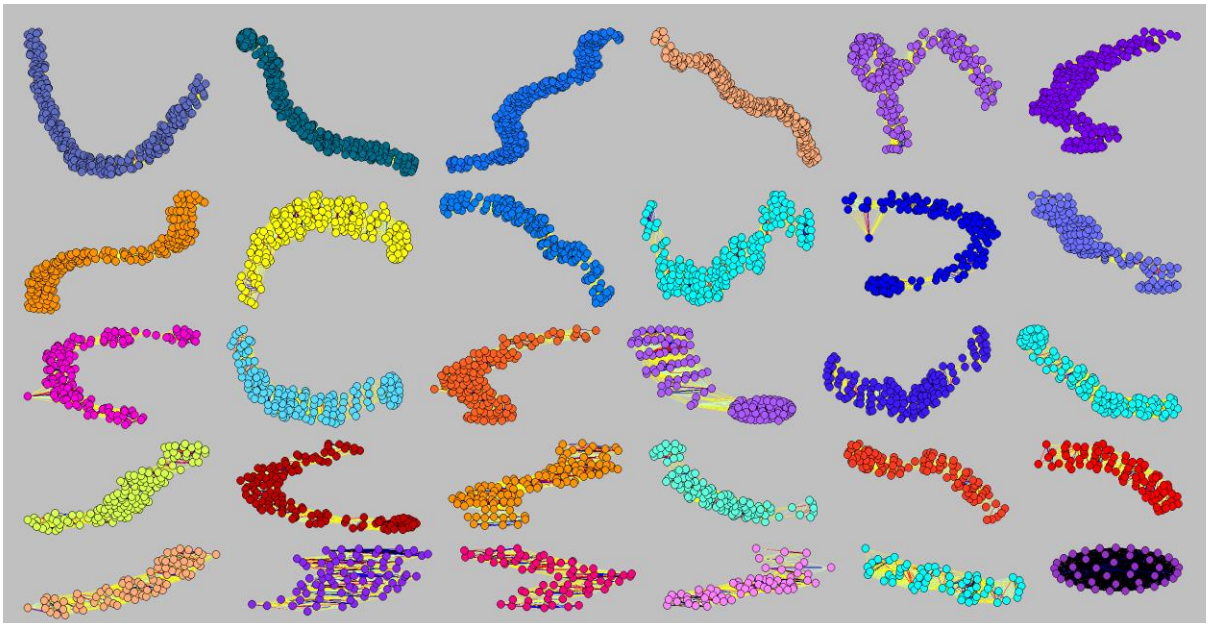


Figure 21. Subdivided linkage groups into linearly structured components (30 linkage groups)

Based on MST of these linkage groups we selected and saved skeleton markers. Based on ordering of skeleton markers within longest path of MST we constructed genetic map. We compared resulted genetic map with one built based on R program. We observed that our map was shorter (about on 10%) and contained more markers. We observed ordering in our map with some difference.

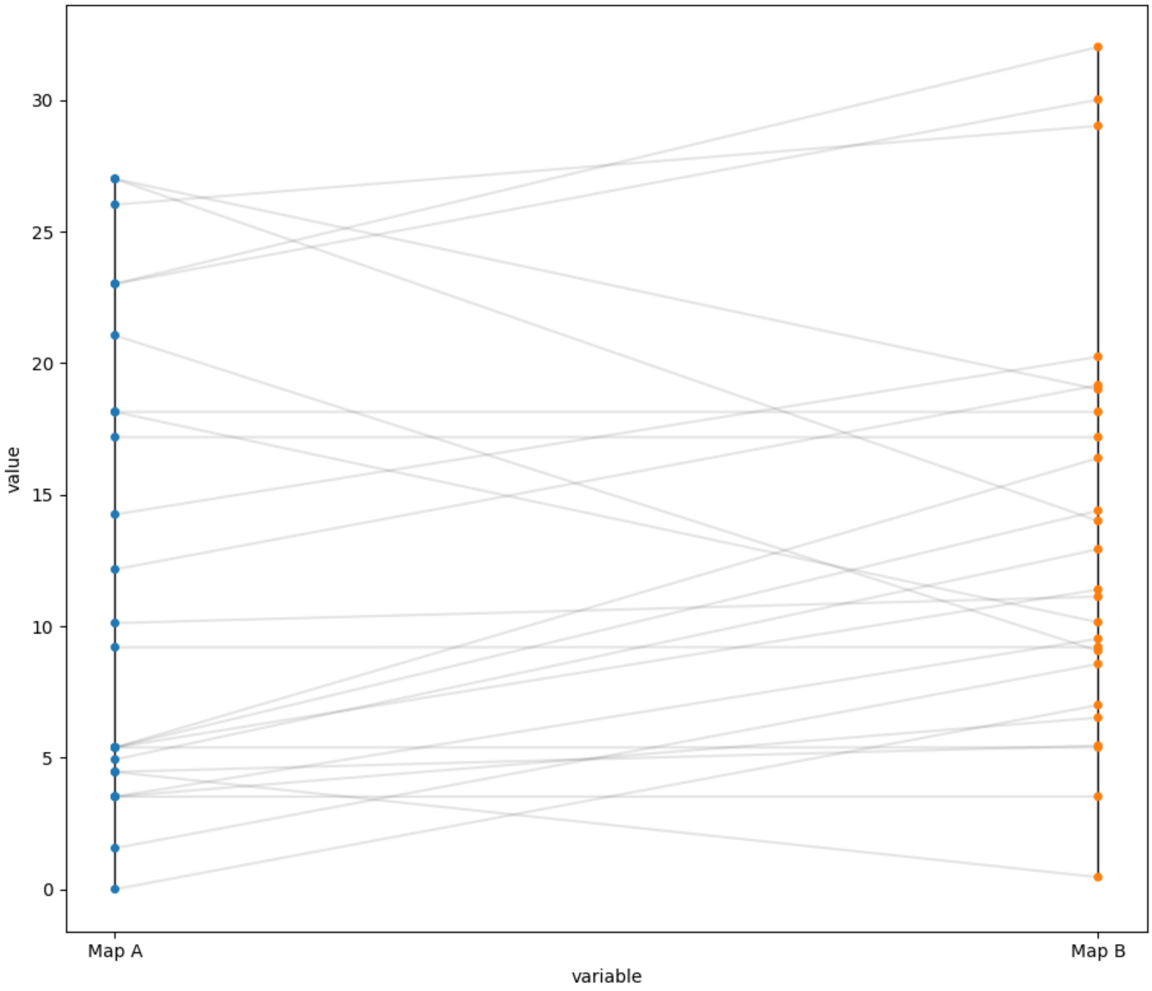
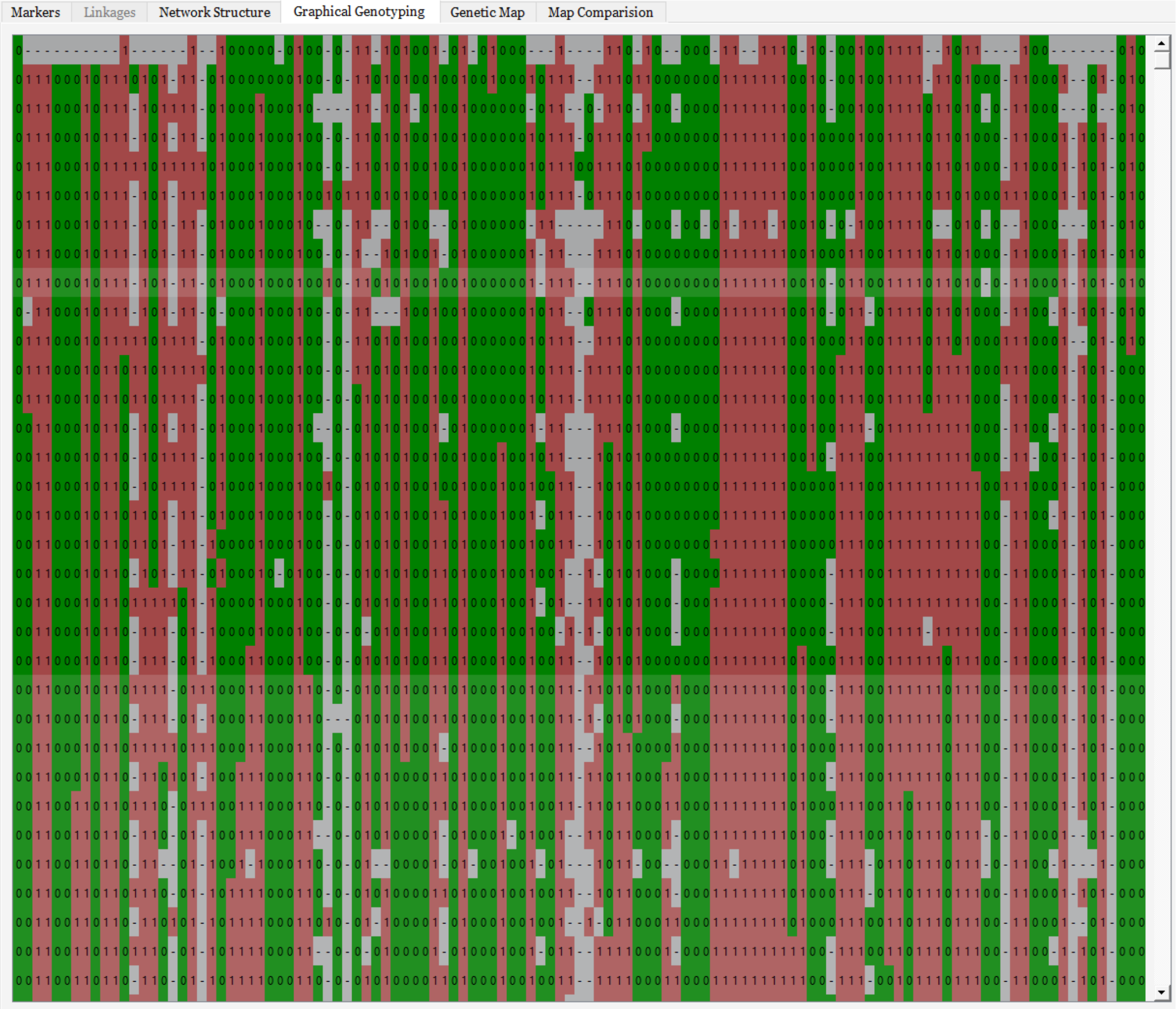


Figure 22. Comparison of differenced between our map and another created by another software.

We also illustrated correctness of our ordering of markers by graphical genotyping



```
0----------1------1--100000-0100-0-11-101001-01-01000---1----110-10--000-11--1110-10-001001111--1011----100-------010
0111000101110101-11-0100000000100-0-11010100100100100010111--111011000000011111110010-001001111-1101000-110001--01-010
011100010111-101111-01000100010----11-101-01001000000-011--0-110-100-000011111110010-001001111011010-0-11000---0--010
011100010111-101-11-010001000100-0-11010100100100000010111-01110110000000111111100100001001111011010000-110001-101-010
01110001011111011111010001000100-0-11010100100100000010111001110100000000111111100100001001111011010000-110001-101-010
011100010111-101-11101000100010010111010100100100000010111-0111010000000011111110010000100111101101000111 0001-101-010
011100010111-101-11-01000100010--0-11--0100--01000000-11-----110-000-00-01-111-10010-0-10011110--010-0--1000---01-010
011100010111-101-11-010001000100-0-1--101001-010000001-11---1110100000000011111110010001100111101101000-110001-101-010
011100010111-101-11-01000100010010-1101010010010000001-111--1110100000000011111110010-011001111011010-0-110001-101-010
0-1100010111-101-11-0-0001000100-0-11---10010010000001011--011101000-000011111110010-011-0111101101000-1100-1-101-010
0111000101111101111-010001000100-0-11010100100100000010111--11101000000000111111100100011001111011010001110001--01-010
01110001011011011111010001000100-0-11010100100100000010111-111101000000000111111100100111001111011110001110001-101-000
0111000101101101111-010001000100-0-01010100100100000010111-111101000000000111111100100111001111011110000-110001-101-000
001100010110-101-11-01000100010--0-010101001-010000001-11---11101000-0000111111100100111-0111111111000-1100-1-101-000
001100010110-101111-010001000100-0-0101010010010001001011---1010100000000011111110010-11100111111111000-11-001-101-000
001100010110-101111-01000100010010-01010100100100010010011--10101000000000111111100000111001111111111001110001-101-000
0011000101101101-11-010001000100-0-0101010011010001001-011--1010100000000011111110000011100111111111100-1100-1-101-000
0011000101101101-11-100001000100-0-01010100110100010010011--1010100000000111111110000011100111111111100-110001-101-000
001100010110-101-11-0100010-0100-0-0101010011010001001001--1-0101000-000011111110000-11100111111111100-110001-101-000
0011000101101111101-100001000100-0-0101010011010001001-01--110101000-000111111110000-11100111111111100-110001-101-000
001100010110-111-01-100001000100-0-0-0101001101000100100-1-1-0101000-000111111110000-111001111-1111100-110001-101-000
001100010110-111-01-100011000100-0-01010100110100010010011--1010100000000111111110100011100111111011100-110001-101-000
0011000101101111-011100011000110-0-01010100110100010010011-1101010001000111111110100-11100111111011100-110001-101-000
001100010110-111-01-100011000110---01010100110100010010011-1-0101000-000111111110100-11100111111011100-110001-101-000
00110001011011111011100011000110-0-010101001-0100010010011--101100001000111111110100011100111111011100-110001-101-000
001100010110-110101-100111000110-0-01010000110100010010011-1101100011000111111110100-11100111111011100-110001-101-000
0011001101101110-011100111000110-0-01010000110100010010011-1101100011000111111110100011100110111011100-110001-101-000
001100110110-110-01-10011100011--0-010100001-010001-01001--110110001-000111111110100-111001101110111-0-110001--01-000
001100110110-11--01-1001-1000110-0-01--00001-01-001001-01---1011-00--00011-111110100-111-011011101111-0-1100-1---1-000
0011001101101110-01-101111000110-0-01010000110100010010011--10110001-00011111111010001000111-0110111011100-110001-101-000
001100110110-110101-10111100011010-01-100001-010001001001--1-01100011000111111110100011100110111011100-110001--01-000
0011001101101110-01-10111100011--0-0-0100001-010001001-011--11110001-0001111111111100-11100110111011100-1100-1-101-000
001100110110-110-01-101111000110-0-01010000110100010010011--111100011000111111111110 0-111-0010111011100-110001-101-000
```

Figure 23. Genotypes after performing swaps for presumably false data providing a much more long vertical lines of same color, improving genotypes quality to presumably as expected. Swaps are based if the number of differences is bigger than the minimum number of 0s or number of 1s

Conclusions:

In our software we successfully managed implement a relatively user-friendly interface for such complicated software, providing a detailed output for the user across the interface. As it excels at dealing with low quality and huge number of markers despite hardware limitations which might increase load time and data processing, but still successfully manage to provide promising and useful results when compared to already existing software in the market with the ability to provide data processing, genotyping, graphic, clustering all in one package.

References:

[1] https://www.genome.gov/genetics-glossary/Centimorgan

[2] https://www.genome.gov/genetics-glossary/Base-Pair

[3] The Human Genome: An Introduction 100-9 - February 2001

[4] An Introduction to Genetic Analysis. 7th edition.

[5] The Genetics of Cancer - Dr. Jay D. Hunt, III

[6] Molecular Level of Genetics

[7] de Vicente C, Fulton T (2003). Molecular Marker Learning Modules. IPGRI, Rome, Italy and Institute for Genetic Diversity, Ithaca, New York, USA.

[8] https://en.wikipedia.org/wiki/Genetic_linkage

[9] J.B.S.Haldane (1919) On Genetic Map Functions

[10] A simple models for recombination Haldane

[11] Genetic Mapping and DNA sequencing Michael S. Waterman, Terry Speed 1996

[12] Kosambi and the Genetic Mapping Function K K Vinod June 2011

[13] Analysis of genetic association using hierarchical clustering and cluster validation indices, Inti A.Pagnuco Juan I.Pastore Guillermo Abras 2017

[14] The Traveling Salesman Problem, Combinatorial Optimization

[15] Genetic Markers, Trait Mapping and Marker-Assisted Selection

[16] Linkage map construction involving a reciprocal translocation, Theor Appl Genet. 2011 Mar;

[17] Genetic Polymorphism in Varietal Identification and Genetic Improvement, M Soller, J.S Beckham 1983

[18] Estimating the effects of population size and type on the accuracy of genetic maps Genet. Mol. Biol. vol.29 no.1 São Paulo 2006

[19] A Multi-Population Consensus Genetic Map Reveals Inconsistent Marker Order among Maps Likely Attributed to Structural Variations in the Apple Genome

[20] Backcross Breeding, Muhammad Jakir Hossain December 2017

[21] High-density genetic map construction and QTL mapping Xiao Zhang,Guo-yun Wang April 2019

[22] LTC: a novel algorithm to improve the efficiency of contig assembly for physical mapping in complex genomes

[23] Travelling Salesman Problem, Holger H. Hoos, Thomas Stützle, in Stochastic Local Search, 2005

[24] Dynamic Programming using Brute Force Algorithm for a Traveling Salesman Problem, Emmanuel Awuni Kolog 2015

[25] Algorithms 4th edition, Robert Sedgewick and Kevin Wayne

[26] https://www.geeksforgeeks.org/branch-and-bound-algorithm/

[27] Genetic algorithms for modelling and optimisation, JohnMcCall

[28] MultiQTL software tools for genome mapping

[29] Concorde for the symmetric traveling salesman problem and network optimization problems.

[30] MSTmap for constructing genetic linkage maps efficiently and accurately.

[31] http://www.geneious.com/