BIOINFORMATICS

Vol. 27 ISMB 2011, pages i257–i265
doi:10.1093/bioinformatics/btr224

Mapping ancestral genomes with massive gene loss: A matrix sandwich problem

Haris Gavranović[1,*], Cedric Chauve[2], Jérôme Salse[3] and Eric Tannier[4,*]

[1]Faculty of Natural Sciences, University of Sarajevo, Bosnia and Herzegovina, [2]Department of Mathematics, Simon Fraser University, Burnaby (BC), Canada, [3]INRA, UMR 1095, Laboratoire Génétique, Diversité et Ecophysiologie des Céréales, 234 avenue du Brézet, 63100 Clermont Ferrand and [4]INRIA Grenoble Rhône-Alpes, LBBE, UMR CNRS 5558, Université de Lyon 1, France

ABSTRACT

Motivation: Ancestral genomes provide a better way to understand the structural evolution of genomes than the simple comparison of extant genomes. Most ancestral genome reconstruction methods rely on universal markers, that is, homologous families of DNA segments present in exactly one exemplar in every considered species. Complex histories of genes or other markers, undergoing duplications and losses, are rarely taken into account. It follows that some ancestors are inaccessible by these methods, such as the proto–monocotyledon whose evolution involved massive gene loss following a whole genome duplication.

Results: We propose a mapping approach based on the combinatorial notion of 'sandwich consecutive ones matrix', which explicitly takes gene losses into account. We introduce combinatorial optimization problems related to this concept, and propose a heuristic solver and a lower bound on the optimal solution. We use these results to propose a configuration for the proto-chromosomes of the monocot ancestor, and study the accuracy of this configuration. We also use our method to reconstruct the ancestral boreoeutherian genomes, which illustrates that the framework we propose is not specific to plant paleogenomics but is adapted to reconstruct any ancestral genome from extant genomes with heterogeneous marker content.

Availability: Upon request to the authors.

Contact: haris.gavranovic@gmail.com; eric.tannier@inria.fr

1 INTRODUCTION

Mapping ancestral genomes consists in ordering ancestral markers into chromosomes, according to the organization of the descendants of these markers in sequenced extant genomes. In the absence of a good model for genome structural evolution, mapping techniques for ancestral genomes, introduced by Bergeron *et al.* (2004), have given the most reliable ancestral configurations on animals (Chauve and Tannier, 2008; Ma *et al.*, 2006; Ouangraoua *et al.*, 2009), yeast (Bertrand *et al.*, 2010; Chauve *et al.*, 2010a; Tannier, 2009), or plant genomes (Murat *et al.*, 2010), and even on a wide eukaryote dataset (Muffato, 2010; Muffato *et al.*, 2010). These works also raised new methodological issues and stimulated a recent stream of algorithmic studies related to genome mapping (Adam *et al.*, 2007; Chauve *et al.*, 2010b, 2009; Dom, 2009; Dom *et al.*, 2010; Manuch and Patterson, 2010; Stoye and Wittler, 2009; Wittler and Stoye, 2010), which had taken the back seat with the development of massive genome sequencing.

The general principle of ancestral genome mapping is:

(i) Define ancestral genome markers, constructed either from homologous gene families or from aligned chromosome segments;

(ii) Infer, from the structure of extant genomes, a collection of relations between ancestral markers which are believed to be ancestral; and

(iii) Assemble this collection into an ancestral genome.

The relations between ancestral markers can take several forms, such as adjacency or distance between pairs of markers, or contiguity/synteny of a subset of markers for example. The combinatorial nature of these relations defines the abstract representation of the ancestral genome, from totally ordered proto–chromosomal segments (Ma *et al.*, 2006; Muffato *et al.*, 2010) to contiguous ancestral regions (Chauve and Tannier, 2008; Ouangraoua *et al.*, 2009) and ancestral linkage groups (Chauve *et al.*, 2010b).

Up to now, most published methods require unique and universal ancestral markers, that is, each ancestral marker has exactly one descendant in every considered extant genome. This constraint, common to many genome–mapping methods (Bergeron *et al.*, 2004; Chauve *et al.*, 2010a; Chauve and Tannier, 2008; Ma *et al.*, 2006; Ouangraoua *et al.*, 2009; Tannier, 2009) and genome rearrangement studies (Alekseyev and Pevzner, 2009; Zhao and Bourque, 2009), results, in general, in more tractable algorithmic problems for the assembly phase. Recently, several works tried to account for the possibly complex history of markers, by integrating whole genome duplication and gene loss either at the level of ancestral markers definition (Chauve *et al.*, 2010a; Murat *et al.*, 2010; Tannier, 2009), or in the whole mapping process, allowing duplicated markers arising from a whole genome duplication (Ma *et al.*, 2008), or genes with duplications and losses (Bertrand *et al.*, 2010; Muffato, 2010; Muffato *et al.*, 2010). In these works, either a backbone of universal makers is used, or only adjacencies between genes were considered. It means that gene loss is neglected, and is expected to produce a reasonable amount of noise in the assembly phase.

The above assumptions are not appropriate anymore if there is a highly heterogeneous marker content within the extant descendants considered to reconstruct an ancestral genome. For example, in the early monocotyledon evolution, a whole genome duplication is believed to have occurred (Abrouk *et al.*, 2010; Salse *et al.*, 2009), followed by numerous gene losses, representing

*To whom correspondence should be addressed a fundamental evolutionary mechanism for these genomes. Gene order in this ancestor is accessible only by the extant relations between paralogous genes that have been kept in two copies, either in the same species or in two different ones. But at this evolutionary distance, only 10% of the genes are of this kind. Universal markers are almost absent, preventing the use of all existing methods except the ones of Muffato (2010) and Bertrand *et al.* (2010). However, adjacencies that can be inferred between ancestral markers are also very sparse, preventing the use of the method of Bertrand *et al.* (2010). Finally, a method using distances and a reduction toward a Traveling Salesman Problem (TSP) is described in the PhD thesis of Muffato (2010), and is close to one of the heuristic principle we describe here. Among our contributions, here is the formalization of this problem and the explicit integration of gene losses in the process.

We propose a solution in the form of a variant of the consecutive ones problem, which was also used in physical mapping of chromosomes and adapted to ancestral mapping (Chauve *et al.*, 2010a; Chauve and Tannier, 2008; Ma *et al.*, 2006; Ouangraoua *et al.*, 2009; Tannier, 2009). We generalize the consecutive ones framework, which itself extends the methods based on adjacencies (Bertrand *et al.*, 2010; Ma *et al.*, 2006; Muffato *et al.*, 2010). The consecutive ones problem is classically used in the following way (see Chauve and Tannier, 2008 for a detailed description). A binary matrix is built, whose columns are the ancestral markers and rows represent groups of markers that are believed to be contiguous in the ancestral genome. It is a binary matrix: there is a 1 in an entry if the marker belongs to the group, and 0 otherwise. Then the assembly phase infers an order of the columns (i.e. markers) such that on all rows the entry ones are consecutive. If such an order, which satisfies the consecutive ones property, does not exist, then a combinatorial optimization approach is used, such as extracting a maximum subset of rows or flipping a minimum number of entries. Groups of markers defining the rows of the matrix are obtained from the comparison of all pairs of extant genomes whose evolutionary path contains the ancestor of interest.

The framework described above can be extended to handle the fact that a marker can be missing when comparing a pair of extant genomes, because of gene loss for example. It simply calls for a third type of entry: if a marker is missing, then it is not possible to say if it belongs or not to a given group of markers, so we may mark it with an entry X in the matrix row associated to this group of markers. In the assembly phase, it can play the role of a 1 or of a 0. The assembly problem is now to find an order of the columns such that on all rows, there is no entry 0 between two entries 1, or, if such an order does not exist, to approximate this property while optimizing a given combinatorial criterion. Note that in the absence of X entries, the problem is equivalent to the consecutive ones described above. This problem was introduced by Golumbic and Wassermann (1998) and Golumbic (1998), under the names ‘sandwich interval hypergraph’ or ‘sandwich consecutive ones matrix’. It was proved that deciding if the columns of a matrix with entries 0, 1 or X (a *ternary matrix*) can be ordered such that no 0 is between two 1s is NP-complete and the possibility to use this concept in physical mapping to account for missing or uncertain data was already mentioned in these articles.

In Section 2, we describe several computational results for the problem of ordering the columns of a ternary matrix. We define combinatorial criteria associated to matrices that do not have the sandwich consecutive ones property, a heuristic based on the technique of partition refinement, a reduction to the TSP, a local search principle and a lower bound that is used to assess the quality of the solutions computed with these methods. Next we apply these algorithmic results to the to reconstruct the gene order of the protochromosomes of the monocotyledon angiosperms before they underwent a whole genome duplication in Section 3.[1] We discuss the robustness of the results and the multiplicity of solutions in Section 3.3. The framework we present is not restricted to plant genomes. It generalizes several of the previous methods, accounting for gene loss as an additional feature. It can be used for any ancestral genome reconstruction. We demonstrate that former results, like the boreoeutherian ancestor, can be retrieved with this new framework with a good accuracy and an increased coverage (Section 4).

2 THE CONSECUTIVE ONE MATRIX SANDWICH PROBLEM

2.1 Problem definition

A ternary matrix M is a matrix with n columns and m rows, each entry being equal to 0, 1 or X. A binary matrix is a ternary matrix without X entries. In a ternary matrix M, an entry $M_{ij} = 0$ is called a bad zero if there exist $a < j$ and $b > j$ such that $M_{ia} = 1$ and $M_{ib} = 1$. A row j is called a bad row if it has a bad zero. Two matrices are equivalent if one can be obtained from the other by a permutation of its columns. A ternary matrix M has the sandwich consecutive ones property (SC1P) if it is equivalent to a matrix with no bad zero. If M is binary and SC1P, then it has the consecutive ones property (C1P).

Typically, we use a ternary matrix M to represent a set of features of an ancestral genome of interest: columns represent ancestral markers (such as ancestral genes for example), and each row represents a group of markers that are believed to be contiguous in this ancestral genome, with all columns with an entry 1 belonging to this group and possibly some columns with an entry X. The goal of the assembly phase in inferring an ancestral genome map is to order the columns of M to represent a possible order of the markers along the proto-chromosome of the ancestral genome. Ideally one would like to find a column order such that M has the SC1P. In practice, it happens, due to convergent evolution, errors in gene annotation or in homology assignment, that a matrix does not have the SC1P, that is, there is no permutation of the columns such that in each row, no zero entry is between two ones (Fig. 1).

The case when a binary matrix does not have the consecutive ones property was the subject of several theoretical and experimental studies (e.g. Chauve *et al.*, 2010b; Dom *et al.*, 2010; Garriga *et al.*, 2008). Different problems arise such as: and computing the largest C1P sub-matrix; and computing the permutation of columns (and rows) that produce the matrix closest to a C1P matrix; computing the minimal number of elements which can be flipped to obtain a C1P matrix. These define optimization problems, most of them NP-hard. Here we use their counterparts in the sandwich problem. A first natural function is defined as the number of bad rows in

[1] A preliminary version of these results is integrated to a large paleogenomics study of cereals (Murat *et al.*, 2010). It was obtained by the partition refinement heuristic, which was not described in Murat *et al.* (2010). All other methodological developments (lower bound, proof of optimality of the solutions, local search, reduction to TSP, robustness study), and the generalization to mammalian genomes, are new.

	a	b	c	d
	7	10	11	12
13	1	1	0	1
17	1	1	0	0
42	X	1	1	0

	a	b	c	d	e	f
	1	6	7	8	12	13
4	1	1	1	0	X	0
12	0	1	1	1	X	1
13	X	X	1	1	0	1
16	0	1	1	1	X	0
43	X	X	0	1	1	0

Fig. 1. Two sub-matrices of the matrix displayed in Figure 6. (Left) Sub-matrix defined by rows (13, 17 and 42) and columns (7, 10, 11 and 12). This matrix has the SC1P, with columns order *dabc*. (Right) Sub-matrix defined by rows (14, 12, 13, 16 and 43) and columns (1, 6, 7, 8, 12 and 13). This matrix does not have the SC1P (Fig. 3). The existence of this sub-matrix proves the optimality of the solution in the Figure 6, because there is only one bad row in the solution.

any equivalent matrix. The problem is then to delete the minimum number of rows so that the remaining matrix has the SC1P. We denote the value of this function for a permutation π on a matrix M by $\lambda_{\text{COS-R}}({}^{\pi}M)$. The problem generalizes the 'path covering problem', used by Ma *et al.* (2006), which is itself a generalization of the Hamiltonian cycle problem. We can also define several objective functions in terms of bad zeros: minimum number of bad zeros among equivalent matrices, $[\mu_{\text{tot}}({}^{\pi}M)]$, maximal number of bad zeros in a row, number of rows having at least k bad zeros for example.

We consider here linear combinations of these functions while creating optimization priority order among them. We ran the local search of Section 2.4 with a function $C\lambda_{\text{COS-R}}({}^{\pi}M)+\mu_{\text{tot}}({}^{\pi}M)$ where C is a big constant. It means that we first aim at minimizing the number of bad rows, and secondarily to integrate bad rows with a minimum number of bad zeros.

We are not aware of any software or even described algorithm attempting the resolution of a consecutive ones sandwich matrix problem. We present below several techniques, a lower bound and a software to solve it and assess the qualities of the solutions. Our solver is based on

- A heuristic based on a partition refinement algorithm to decide if a binary matrix has the consecutive ones property (McConnell, 2004) (Section 2.2).
- A reduction to the TSP (Section 2.3).
- A local search to find a local optimum close to the results found by the two above methods (Section 2.4).

We also describe a lower bound on the number of bad rows in a matrix that does not have the sandwich consecutive ones property, based on the certificate described in McConnell (2004) (Section 2.5). It is used to assess the quality and, in some cases, to prove optimality of the solutions obtained.

2.2 A heuristic based on partition refinement

The heuristic we describe now is based on the C1P matrix recognition algorithm described in McConnell (2004), and relies on the very general algorithmic tool of partition refinement. It is a generalization of this algorithm, in the sense that if the instance is a binary matrix with the C1P, it performs the partition refinement just as described in McConnell (2004).

First, rows of the matrix are partitioned into connected components of the overlap graph as follows: vertices are rows and two rows define an edge if they overlap, i.e. if they have some common columns with an entry 1, but none is contained in the other.

Then, for each component, rows are ordered according to a breadth first search, and processed according to this order. The goal is to partition the columns spanned by the component into a totally ordered set $\{X_1,\dots,X_k\}$. Every set X_i (a set of columns) is unordered but the partition itself is. Components are processed independently.

During the processing of a component, a given column can be assigned to several $X_i's$, which is the main difference with the classical partition refinement algorithm used to decide the C1P. So the structure which is maintained is an ordered family $\{X_1,\dots,X_i\}$ plus a set X_0 of unassigned columns, and a function from the columns to a sub-family of $X_0,\dots,X_i$. The image of a column is called its possible assignments (in the algorithm of McConnell (2004), there is only one possible assignment, and several assignments are used only if X entries are encountered in the matrix). Below, we still use the terminology partition for the intermediate $X_i's$ even if, formally, they do form a partition of the columns only at the end of the process.

Initialization: at first, all columns spanned by the component are assigned to X_0. The first row r_1 is treated the ones in r_1 are assigned to X_1, the zeros to X_0 and the X entries to both X_0 and X_1.

Induction: then, for a row r_j, processed after rows $r_1,\dots,r_{j-1}$, assume the current columns partition is $\{X_1,\dots,X_i\}$. Let $\{X_a,\dots,X_b\}$ be the largest interval of the partition such that (1) for every column with a 1 in r_j, one of its possible assignments is in either X_0 or $X_a,\dots,X_b$, and (2) for every column with a 0 in r_j, some of its possible assignments are outside $\{X_{a+1},\dots,X_{b-1}\}$.

If such interval does not exist, skip row r_j and process the following row. Else add two sets: $X_{a'}$ before X_a and $X_{b'}$ after X_b. For each column c, (1) if it has a 0 in r_j and a possible assignment in X_a (respectively X_b), replace the assignment of c to X_a (respectively X_b) by an assignment to $X_{a'}$ (respectively $X_{b'}$), (2) if it has a X in r_j and a possible assignment in X_a (respectively X_b), add $X_{a'}$ (respectively $X_{b'}$) to the possible assignments of c, and (3) if it has a 1 in r_j and some possible assignments in $\{X_a,\dots,X_b\}$, then remove all its other possible assignments. Finally, remove empty sets.

If there is a column with a 1 in r_j and without a possible assignment within $\{X_a,\dots,X_b\}$, then this column is currently assigned to X_0. In this case, either $a=1$ or $b=i$, as the rows are processed accorded to a breadth first search of the current component. If $a=1$ (respectively $b=i$), then $X_{a'}$ (respectively $X_{b'}$) is empty (in the opposite case skip row r_j and process to the following row). Let $X_{a'}$ (respectively $X_{b'}$) is the set containing all columns with a 1 in r_j but without an assignment within the sets $\{X_1,\dots,X_i\}$. Insert it before X_a (respectively after X_b). If a column has a X in r_j and a possible assignment to X_0, then we add $X_{a'}$ (respectively $X_{b'}$) as a possible assignment for this column.

The result of this algorithm is then a partition of the columns into a totally ordered set $\{X_1,\dots,X_k\}$. From a theoretical point of view, this heuristic has the important property that, if M does not contain any entry X, this partition is the one computed by the algorithm of McConnell (2004) to decide the C1P. If M does not satisfy the $C1PX$, then the columns order defined by this partition induces bad rows or bad zeros.

2.3 Reduction to the TSP

It is well known that some variations of C1P can naturally be reduced to the TSP. We describe here such an approach, bearing some similarity with the method of Muffato (2010) applied to teleost fishes: (i) construct a complete graph $G=(V,E)$ where the set of vertices corresponds to the columns of the matrix and (ii) assign to every edges an appropriate cost/weight [Hamming distance, Jaccard coefficient, see Garriga *et al.* (2008) for example]. A Hamiltonian path in this graph represents a permutation of the columns, i.e. an order of the markers along ancestral chromosomes, and the shortest Hamiltonian path represents a solution tending to optimize the objective function defined by the cost model.

Here, we define a modified Hamming distance δ_H and modified Multiply Transpose distance δ_{MT} for two columns a and b of a ternary matrix M, parametrized by a real number α, as follows:

$$\delta_{\mathrm{H}}(a,b)=\sum_{\text{rows of } M}\begin{cases}0 & 00\\ 0 & 11\\ \alpha & 0X\\ 0 & 1X\\ 1 & 10\end{cases}$$

$$\delta_{\mathrm{MT}}(a,b)=10000-\sum_{\text{rows of } M}\begin{cases}0 & 00\\ 1000 & 11\\ 0 & 0X\\ \alpha\cdot 10 & 1X\\ \alpha^2/10 & XX\\ 0 & 10\end{cases}$$

To solve the obtained TSP instances, we use the publicly available TSP solver Concorde, with Ilog CPLEX 11.0. Results of calculation are reported in Table 2. We tried both distances with several parameters, and chose the best solution. Except for huge matrices, the solution is not very sensitive to the parameter variation.

2.4 Local search

To improve the solutions obtained by one of the heuristics described above, we devised a simple local search that modifies the order of the matrix columns. The five basic moves we considered here are:

- move one column from a position to another one,
- move a set of consecutive columns to another position,
- swap the position of two columns,
- reverse the order of a set of consecutive columns, and
- move and Reverse, i.e. move a set of consecutive columns to another position and in reverse order.

We implemented the local search with an objective function first tending to minimize the number of bad rows and secondarily trying to minimize the number of bad zeros in bad rows.

2.5 Lower bound

To measure the quality of the solutions given by the combination of a constructive heuristic followed by a local search, we define a lower bound for the objective function that counts the number of bad rows in a solution, i.e. a lower bound on the minimal number of rows to remove from the matrix so that, for the remaining rows, there is an order of the columns with no bad row. Our approach is based on the notions of forbidden substructure characterization and incompatibility graph developed by McConnell (2004).

2.5.1 The incompatibility graph and forbidden configurations We first recall the construction of the incompatibility graph. Let a and b be two columns of matrix M and denote by (a,b) the fact that a appears before b in the order of columns. Let G be an undirected graph whose vertices are the elements of $\{(a,b)|a,b \text{ are columns and } a\neq b\}$. Edges of G indicate the incompatibility between column orders with respect to the SC1P.

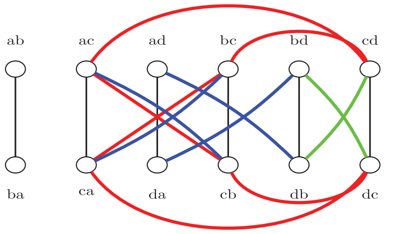


Fig. 2. The incompatibility graph associated to the sub-matrix in Figure 1 (Left). Red edges are associated with row 13, blue edges are associated with row 17 and green edges come from row 42. This graph is bipartite and the bipartition for SC1P is $(A=\{ab,ac,da,bc,db,dc\},\overline{A})$ which also defines the order *dabc*.

- In any given order, (a,b) and (b,a) cannot appear simultaneously. Thus, a first set of edges connects every pair (a,b),(b,a).
- Suppose now that a,b,c are three columns and there exists a row r such that $M_{ra}=M_{rc}=1$ and $M_{rb}=0$. In an order with the SC1P, (a,b) and (b,c) cannot hold both. We, therefore, say that (a,b) and (b,c) are incompatible and define an edge between these two vertices. We associate the row r to this edge; therefore, the graph G can have multi-edges between two pairs of vertices but these edges are distinguished by the associated rows. Note, however, that if any of these value is X, then we can not say anything about incompatibility.

McConnell (2004) proved that the incompatibility graph of a binary matrix is bipartite if and only if the matrix itself has the C1P, which provides a key ingredient for a certificate for the C1P. A similar, although weaker property, also holds for the SC1P.

Property: if the incompatibility graph of a ternary matrix M contains an odd-length cycle, then M does not satisfy the SC1P.

Note that the vertices and edges of an odd-length cycle C_o define a set of columns and rows and, therefore, define a sub-matrix of M, that we call a forbidden configuration and that we denote M_{C_o}, which does not have the SC1P by the property above. Examples of such sub-matrices are given in Figure 1. They are sub-matrices of the matrix obtained from the monocot proto-chromosome A11 shown on Figure 6. The incompatibility graphs for those two sub-matrices are drawn on Figures 2 and 3.

2.5.2 Disjoint forbidden configurations Two odd cycles are called strongly disjoint in the incompatibility graph if they are disjoint in the graph, and if they do not share any row in the matrix. If there exists a set of k odd cycles that are pairwise strongly disjoint odd cycles, this clearly implies that there is at least k bad rows, as each sub-matrix associated with a cycle contains at least one bad row. We then introduce the problem of computing the maximum number of odd cycles that are, pairwise, strongly disjoint, and we describe below an efficient heuristic for this problem.

First, we compute a set of odd cycles as follows. One odd cycle is found by searching the graph, and, in the associated sub-matrix, we flip the zero entries to X. In this way, this odd cycle vanishes

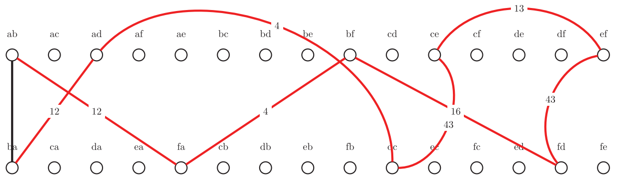


Fig. 3. The incompatibility graph associated with the sub-matrix from Figure 1 (Right). Every edge is labeled with the corresponding row except the edge (ab,ba) which represents an edge of first type. Note that for the cycle we need all rows and all columns from the sub-matrix from the Table 1. This odd-length cycle gives a certificate that the sub-matrix does not have the SC1P.

and we start again the search for another odd cycle, until the graph is bipartite. The set of odd cycles so obtained are still not strongly disjoint. So we find the maximum subset of strongly disjoint ones by solving an independent set problem on the graph which vertices are the odd cycles and two vertices are joint by an edge if they are not strongly disjoint. We solve this problem using an integer programming formulation and solver.

For the matrix M with m columns, the number of vertices in the incompatibility graph is $m\times(m-1)$ but all previous calculations remain valid if we choose the suitable subset of columns, for example the set columns comprising one or several bad zeros and repeat the procedure to cover whole matrix. This is how, if needed, we can reduce the size of the graphs we deal with. The number of odd cycles can be very large, and this leads to rather big integer programs to solve. But all these programs were easily solved for small and medium instances (monocot ones) as it is reported here.

3 RECONSTRUCTING THE ANCESTRAL MONOCOTYLEDON GENOME

Monocotyledons are a branch of angiosperms whose genome has undergone a global duplication at an early stage of its evolution (Salse *et al.*, 2009). We describe here how to define ancestral markers for this genomes, then how to compute ternary matrices representing putative features of the ancestral monocotyledon genome, and finally the result of the computational methods described in Section 2 on these matrices.

3.1 Ancestral markers and ternary matrices

Two paralogous chromosomes or genes arising from a whole genome duplication are called ohnologous. Ohnologous chromosomes and chromosome segments were identified in Murat *et al.* (2010), and we use the gene homologies computed in the same study. For example, Rice chromosomes 1 and 5 are ohnologous, as well as Sorghum chromosomes 3 and 9. As all four arise from the whole genome duplication (Rice 1 is orthologous to Sorghum 3 and Rice 5 to Sorghum 9, due to a speciation posterior to the whole genome duplication), Rice 1 and Sorghum 9 are ohnologous, as well as Rice 5 and Sorghum 3. This gives four ohnologous relations on these four chromosomes summarized in Figure 4, where ohnology relationships between genes are drawn with gene positions in chromosomes. In the present work, we define a relation between genes as follows: an ohnologous pair of genes is a pair of paralogous genes that are located on two ohnologous segments. All ohnologous pairs on Rice chromosomes 1 and 5 and Sorghum chromosomes 3 and 9 are drawn in Figure 4. Genes are grouped into families defined by the transitive closure of the relation, and each family defines an ancestral marker.

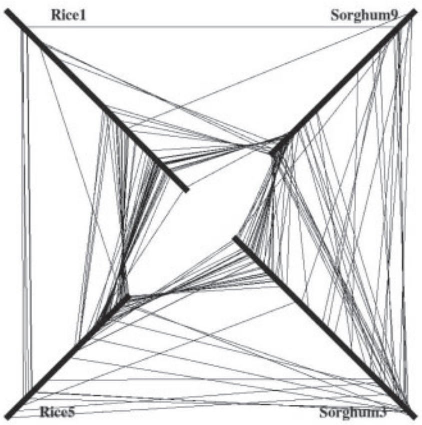


Fig. 4. Four ohnologous chromosomes (bold lines) and the ohnolog pairs of genes (thin lines).

We then define a matrix M, whose columns are the ancestral markers and rows are defined in terms of adjacencies and common intervals. Then for each pair of ohnologous segments A and B, we computed common intervals (Chauve and Tannier, 2008) on the set of ancestral markers which have descendants in both segments. For each common interval I, a row of M is constructed as follows:

- there is a 1 in column i if the ancestral marker i is in I,
- there is a 0 in column i if the ancestral marker i has descendants in A and B but is not in I, and
- there is an X in column i if the ancestral marker i has no descendant in either A, or B, or both.

This defines the monocot ternary matrix. A toy example is given in Figure 5, based on two segmental ohnologous relations.

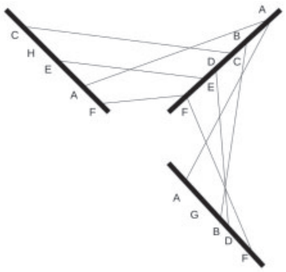
Common intervals are sets of ancestral genes which descendants are seen contiguous in two branches arising from the whole genome duplication, so they are believed to define a set of contiguous markers in the ancestor. In the example of Figure 5, moving column A into the first position results in a matrix where no entry 0 is located between two entries 1, which means it satisfies the sandwich consecutive ones property.

We applied the technique described above to define five ternary matrices defining 5–7 proto-chromosomes of the monocotyledon ancestral genome (Murat *et al.*, 2010) (proto-chromosomes A4, A5, A7, A8 and A11). See Table 1.

3.2 Proto-chromosomes of the monocot ancestor

Figures 6 and 7 give examples of the shape of the instances and of the solutions. The shown ternary matrices represent the input used to compute proto–chromosomes A11 and A5 (Murat *et al.*, 2010). Entries 1 are red, 0 are blue and X are green. All rows are represented, even the bad ones. Red segments represent common intervals. For A11, only 35 ancestral genes were considered, whereas 120 genes were considered in A5.

A11 has only one bad row and as shown in the previous section the matrix does not have the SC1P. So this solution is optimal in terms



	E	C	B	D	H	A	G	F
	1	1	X	X	X	0	X	0
	1	1	X	X	X	1	X	0
	1	1	X	X	X	1	X	1
	X	X	1	0	X	1	X	0
	X	X	1	1	X	0	X	0
	X	X	0	1	X	0	X	1

Fig. 5. (Top) Three segments sharing two ohnologous relations, together with 8 ohnologous pairs of genes. (Bottom) The ternary matrix obtained by computing all common intervals. The three first rows encode the common intervals between the two upper segments, and the last three rows encode the common intervals between the two right segments.

Table 1. Characteristics of the five considered instances: name, number of rows, number of columns, ratio of 1s and Xs in the matrix

Instances				
Name	No. of rows	No. of columns	%1	%X
A4	748	139	5.79	59.02
A5	548	120	7.43	42.67
A7	144	164	2.00	55.43
A8	548	48	18.40	30.63
A11	50	35	14.06	34.55

of the number of bad rows, and in the bad row there is a solution with only one bad zero, then optimality is guaranteed also for the number of bad zeros. For A5, the lower bound gives at least 27 bad rows, whereas we find a solution with 34 ones (over a total of 548 rows). Statistics for the other chromosomes[2] are given in Table 2.

To assess what we gained by modeling the result of gene loss instead of not taking it into account and consider it as a possible source of errors in the data, we also computed the solutions by replacing X entries by 0. We used the method of Chauve and Tannier (2008) to solve the problem of the minimum number of rows to discard, in order for the remaining matrix to have the C1P. In the A11 matrix, 11 rows have to be discarded (C1P) instead of only 1 (SC1P). For A5, 351 rows are bad instead of 59. So the noise due to some errors in the data is considerably reduced by paying attention to gene loss.

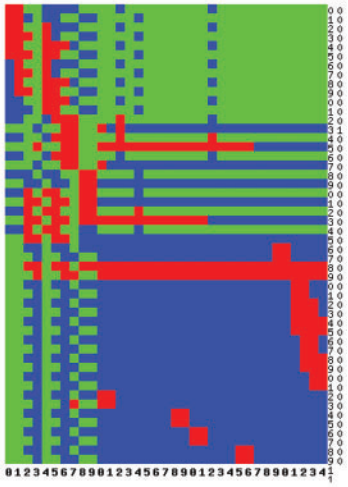


Fig. 6. A possible gene order of proto-chromosome A11 of monocots. There is only one bad row and one bad zero.

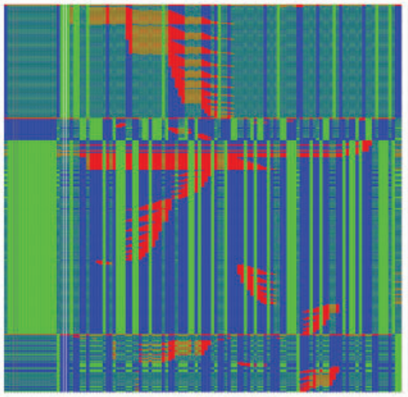


Fig. 7. A possible gene order of proto-chromosome A5 of monocots.

Table 2. Results obtained for two objective functions: number of bad rows and number of bad zeros

Bad rows					Bad zeros			
Name	Mc	TSP	+LS	LB	Mc	TSP	+LS	LB
A4	27	100	11	8	299	1045	57	–
A5	59	170	34	27	749	1859	432	–
A7	**3**	14	3	**3**	**3**	50	3	–
A8	99	161	93	55	514	250	133	–
A11	**1**	3	1	**1**	2	15	1	–

Column Mc describes the results of the heuristic of Section 2.2. It reaches the optimal value for instances $A7$ and $A11$ and consistently provides solutions of quality for other instances. TSP reduction is numerically less competitive but still gives meaningful results. The local search often reaches the reported values even if it starts from a random order of columns but its efficiency is enhanced with good starting solutions from the previous two procedures. Finally, in the LB column we report the obtained lower bounds for the number of bad rows. Bold values point at optimal solutions obtained by heuristics.

[2] A4, A7, A8 as named in Murat *et al.* (2010), even if two of them are split into at two segments in our reconstruction, leading to 5, 6 or 7 proto-chromosomes according to the branch in which the two probable rearrangements occurred.

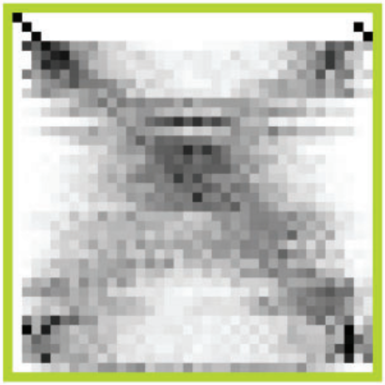


Fig. 8. Representation of 950 good solutions for the A11 proto-chromosome.

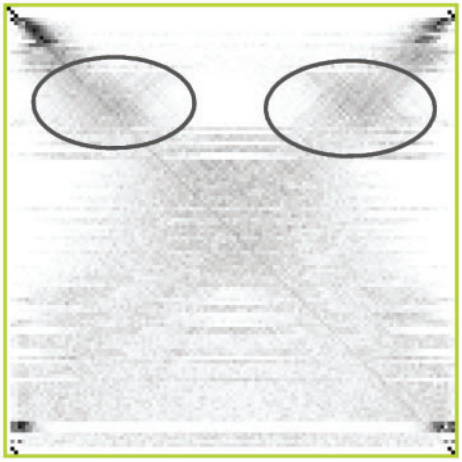


Fig. 9. Representation of 950 good solutions for the A5 proto-chromosome.

3.3 Validation and robustness

Figures 8 and 9 represent a sample of locally optimal solutions obtained by repeatedly applying the program with a randomization (the local search is easily randomized for example, as well as the order in which rows are processed in the partition refinement heuristic). In these figures, the order of the columns we represent is the one of a best found solution, and a gray square at coordinates i,j represents the proportion of solutions in which column i is in position j: the darker the square is, the larger the proportion is.

The 'X' shape of the figures is explained by the symmetrical property of the problem: one order of the columns and its reverse have the same score according to all objective functions. Few columns have a compulsory position. But the distribution of the positions of most columns is far from uniform: there is a clear tendency to stay close to the position in the chosen best solution. In Figure 9, we can see an inversion for which it is impossible to decide whether the ancestral state carries it or not (the small 'X' shapes surrounded by a black ellipse). This is not surprising from the examination of Figure 4, where we can see that Rice chromosome 1 and Sorghum chromosome 3 carry one order on this part, while Rice 5 and Sorghum 9 exhibit the reverse order. So there has been an inversion along one evolutionary branch (the one leading to Rice1/Sorghum3, the other to Rice5/Sorghum9) after the duplication, but there is no way to decide on which one, so we cannot know the ancestral configuration.

4 RECONSTRUCTING THE MAMMALIAN ANCESTOR

The SC1P of ternary matrices generalizes the C1P for binary matrices. So the framework we present is not only a way to reconstruct ancestral genomes with massive gene losses as the monocot one, but also all problems solved by C1P techniques can be *a fortiori* solved in this case, with possibly more accuracy in the presence of unequal gene/marker content in extant species. Indeed, as discussed in Pham and Pevzner (2010) for example, when large datasets of extant genomes are considered, the number of universal markers naturally decreases, due, for example, to lineage–specific rearrangements that split the occurrence of a marker into several smaller genome segment. As a consequence usual methods to infer synteny blocks, either from genes or DNA alignments, are not adapted anymore.

Here we show that the SC1P can be used to handle such problems, and we illustrate this feature by reconstructing the boreoeutherian ancestor, which has been the subject of a large literature (including Chauve and Tannier, 2008; Ma *et al.*, 2006; Muffato *et al.*, 2010) and whose general architecture is mostly agreed upon by both computational and cytogenetics studies.

4.1 Ancestral markers

We used the Pecan alignments from Ensembl Compara (version 58, Paten *et al.*, 2008), which is a set of non-universal homologous markers within 15 amniote genomes, including 12 placental mammals (*Homo sapiens*, *Pan troglodytes*, *Gorilla gorilla*, *Pongo pygmaeus*, *Callithrix jacchus*, *Macaca mulatta*, *Mus musculus*, *Rattus norvegicus*, *Equus caballus*, *Canis familiaris*, *Bos taurus*, *Sus scrofa*, *Monodelphis domestica*, *Gallus gallus* and *Taeniopygia guttata*). Alignments that were colinear in all genomes in which they are present were first grouped, and then alignments of length <100 kb in at least one species were discarded.

Then an ancestral marker is defined by each alignment which has at least an occurence in two extant species whose evolutionary path goes through the boreoeutherian node. This gives 1724 ancestral markers covering 35% of the human genome, whereas there was 990 universal markers (i.e. present in all 15 species) covering 24% of the human genome. So allowing non-universal markers results in a significant improvement of the coverage of extant genomes by ancestral markers.

4.2 Reconstructing proto-chromosomes

A ternary matrix for the boreoeutherian ancestral genome was constructed by performing pairwise comparisons of extant genomes. For each pair of species whose evolutionary path goes though the boreoeutherian ancestor, we computed conserved adjacencies and maximal common intervals on the set of ancestral markers which have a descendant in both genomes. The ternary matrix was then constructed in the same way than for the monocot ancestor, with entries X used to represent markers that were missing (lost) in at

Table 3. Boreoeutherian proto-chromosomal segments, compared to human genome

With bad rows			Without bad rows		
Id	Length	Assoc.	Id	Length	Assoc.
1	111780676	1 19 16	1	111780676	1 19 16
2	95479039	6	2	95479039	6
3	77595574	3 21	3	77595574	3 21
4	74910913	4 8	4	74910913	4 8
5	64738531	15 14	5	64738531	15 14
6	63586057	5	6	63586057	5
7	56922221	7	7	56922221	7
8	55923673	12 22 10	8a	11341365	12 22
			8b	44582308	10
9	55529695	8	9	55529695	8
10	55036330	2	10	55036330	2
11	45149221	12 22	11	45149221	12 22
12	44208695	9	12	44208695	9
13	40880160	X	13	40880160	X
14	40509571	19 11	14a	38723712	11
			14b	1785859	19
15	37823275	13	15	37823275	13
16	35906505	2	16	35906505	2
17	32491882	18	17	32491882	18
18	25119674	20	18	25119674	20
19	18273205	17	19	18273205	17
20	16763121	10	20	16763121	10
21	5083976	7	21	5083976	7
22	2005762	16	22	2005762	16
23	980679	22	23	980679	22
24	771170	2	24	771170	2

Length is the cumulated length of the markers, taken on the human genome, composing the segments. Assoc. represents the chromosomal associations, again for the human genome, i.e. the human chromosomes containing the markers present on each proto-chromosomal segment.

least one of the two considered genomes. This matrix contains 1724 columns and 89 023 rows.

After application of the SC1P solver, the resulting columns order implied 316 (0.3%) bad rows and 2753 bad zeros. Keeping the bad rows in the matrix defines 24 proto-chromosomal segments, while discarding the 316 bad rows split some of these segments into several sub-segments and resulted in 26 proto-chromosomal segments. Characteristics of these proto-chromosomal segments are given in Table 3.

It appears that the number of proto-chromosomes obtained when keeping the bad rows is comparable to what is usually expected for this ancestral genome (Froenicke *et al.*, 2006). Moreover, the chromosomal associations that are observed are globally consistent with previous studies, but for ancestral segments 1 and 14. Segment 14 is clearly chimeric and joins two proto-chromosomes (one corresponding to human chromosome 11 and one corresponding to segments of human chromosome 19). It is interesting to see it is split into two segments when discarding the bad rows. Similar observation does not hold for segment 1, also probably chimeric. Understanding the signal that results in these CAR will be key to refine the SC1P approach. Finally, we can see that segment 8 seems to capture an association between human chromosomes 12, 22 and 10 (although lost when discarding bad rows) that has been described in cytogenetics studies (Froenicke *et al.*, 2006) but never recovered in computational studies based on sequenced genomes until now.

In conclusion, we recover an ancestral genome which fully meets the standards of usual reconstructions, with some additional interesting features and a higher coverage than if only universal markers were used.

5 CONCLUSION

We introduce a general framework for ancestral genome reconstruction by genome mapping, which contains the principles of most former methods and adds the possibility to handle heterogenous genome content and, in particular, gene losses. We apply it to the reconstruction of the pre-duplication monocot ancestor, which is inaccessible to other methods requiring universal genes or conserved adjacencies. We also show that it can be applied to other taxonomic groups as illustrated with the boreoeutherian ancestor, by providing a solution with a higher coverage than if markers were restricted to universal markers.

The solution we describe is based on a classical combinatorial problem, the sandwich consecutive ones matrix. We propose a software based on several algorithmic techniques to solve related optimization problems. We also assess the quality of our solutions by computing a lower bound of the optimal solution, and by representing a large set of locally optimum solutions.

The results we obtain suggest that the framework is well adapted to handle a heterogeneous gene/marker content in paleogenomics studies. This should motivate further investigations on related methodological and algorithmic problems. In order to process large datasets, algorithms will have to be able to handle very large instances, with dozens of thousands of columns and possibly several hundreds of thousand of rows. Currently, we are limited to one order of magnitude below. Moreover, to decrease the number of entries *X* in the ternary matrices, and then the number of bad rows, better models of ancestral markers and common intervals with unequal gene/marker content should be investigated. It is also possible to generalise this framework by explicitly modeling gene duplications, adding the possibility of multiplicities as in Wittler and Stoye (2010). This addition will be the topic of a future work.

Funding: Agence Nationale pour la Recherche (ANR-08-GENM-036-01 and ANR-08-EMER-011-03 to E.T.); NSERC Discovery Grant (to C.C.).

Conflict of Interest: none declared.

REFERENCES

Abrouk,M. *et al*. (2010) Palaeogenomics of plants: synteny-based modelling of extinct ancestors. *Trends Plant Sci.*, **15**, 479–487.

Adam,Z. *et al*. (2007) Common intervals and symmetric difference in a model-free phylogenomics, with an application to streptophyte evolution. *J. Comput. Biol.*, **14**, 436–445.

Alekseyev,M.A. and Pevzner,P.A. (2009) Breakpoint graphs and ancestral genome reconstructions. *Genome Res.*, **19**, 943–957.

Bergeron,A. *et al*. (2004) Reconstructing ancestral gene order using conserved intervals. *Lect. Notes Comput. Sci.*, **3240**, 14–25.

Bertrand,D. *et al*. (2010) Reconstruction of ancestral genome subject to whole genome duplication, speciation, rearrangement and loss. *Lect. Notes Comput. Sci.*, **6293**, 78–89.

Chauve,C. *et al*. (2010a) Yeast ancestral genome reconstructions: the possibilities of computational methods II. *J. Comput. Biol.*, **17**, 1097–1112.

Chauve,C. *et al*. (2010b) Minimal conflicting sets for the consecutive ones property in ancestral genome reconstruction. *J. Comput. Biol.*, **17**, 1167–1181.

Chauve,C. *et al*. (2009) On the gapped consecutive-ones property. *Electron. Notes Discrete Math.*, **34**, 121–125.

Chauve,C. and Tannier,E. (2008) A methodological framework for the reconstruction of contiguous regions of ancestral genomes and its application to mammalian genomes. *PLoS Comput. Biol.*, **4**, e1000234.

Dom,M. (2009) Algorithmic aspects of the consecutive-ones property. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, **98**, 27–59.

Dom,M. *et al*. (2010) Approximation and fixed-parameter algorithms for consecutive ones submatrix problems. *J. Comput. Syst. Sci.*, **76**, 204–221.

Froenicke,L. *et al*. (2006) Are molecular cytogenetics and bioinformatics suggesting diverging models of ancestral mammalian genomes? *Genome Res.*, **16**, 306–310.

Garriga,G.C. *et al*. (2008) Banded structure in binary matrices. In Li,Y. *et al*. (eds), *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24–27, 2008*, ACM, pp. 292–300.

Golumbic,M. and Wassermann,A. (1998) Complexity and algorithms for graph and hypergraph sandwitch problems. *Graphs Combin.*, **14**, 223–239.

Golumbic,M.C. (1998) Matrix sandwich problems. *Linear Algebra Appl.*, **277**, 239–251.

Ma,J. *et al*. (2008) Dupcar: reconstructing contiguous ancestral regions with duplications. *J. Comput. Biol.*, **15**, 1007–1027.

Ma,J. *et al*. (2006) Reconstructing contiguous regions of an ancestral genome. *Genome Res.*, **16**, 1557–1565.

Manuch,J. and Patterson,M. (2010) The complexity of the gapped consecutive-ones property problem for matrices of bounded maximum degree. *Lect. Notes Comput. Sci.*, **6398**, 278–289.

McConnell,R.M. (2004) A certifying algorithm for the consecutive-ones property. In Munro,I.J. (ed) *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11–14, 2004*, SIAM, pp. 768–777.

Muffato,M. (2010) Reconstruction de génomes ancestraux chez les vertébrés. PhD Thesis, Université d'Evry Val d'Essonne.

Muffato,M. *et al*. (2010) Genomicus: a database and a browser to study gene synteny in modern and ancestral genomes. *Bioinformatics*, **26**, 1119–1121.

Murat,F. *et al*. (2010) Ancestral grass karyotype reconstruction unravels new mechanisms of genome shuffling as a source of plant evolution. *Genome Res.*, **20**, 1545–1557.

Ouangraoua,A. *et al*. (2009) Prediction of contiguous ancestral regions in the amniote ancestral genome. *Lect. Notes Comput. Sci.*, **5542**, 173–185.

Paten,B. *et al*. (2008) Enredo and pecan: genome-wide mammalian consistency-based multiple alignment with paralogs. *Genome Res.*, **18**, 1814–1828.

Pham,S.K. and Pevzner,P.A. (2010) Drimm-synteny: decomposing genomes into evolutionary conserved segments. *Bioinformatics*, **26**, 2509–2516.

Salse,J. *et al*. (2009) Reconstruction of monocotelydoneous proto-chromosomes reveals faster evolution in plants than in animals. *Proc. Natl Acad. Sci. USA*, **106**, 14908–14913.

Stoye,J. and Wittler,R. (2009) A unified approach for reconstructing ancient gene clusters. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **6**, 387–400.

Tannier,E. (2009) Yeast ancestral genome reconstruction: the possibilities of computational methods. *Lect. Notes Comput. Sci.*, **5817**, 1–12.

Wittler,R. and Stoye,J. (2010) Consistency of sequence-based gene clusters. *Lect. Notes Comput. Sci.*, **6398**, 252–263.

Zhao,H. and Bourque,G. (2009) Recovering genome rearrangements in the mammalian phylogeny. *Genome Res.*, **19**, 934–942.