THE EFFECT OF MASSIVE GENE LOSS FOLLOWING WHOLE GENOME DUPLICATION ON THE ALGORITHMIC RECONSTRUCTION OF THE ANCESTRAL POPULUS DIPLOID

Chunfang Zheng
Department of Biology, University of Ottawa,
Ottawa, Ontario K1N 6N5, Canada
Email: czh033@uottawa.ca

P. Kerr Wall
Biology Department, Penn State University,
University Park, PA 16802, USA
Email: pkerrwall@psu.edu

Jim Leebens-Mack
Department of Plant Biology, University of Georgia,
Athens, GA 30602, USA
Email: jleebensmack@plantbio.uga.edu

Victor A. Albert
Joint Centre for Bioinformatics, University of Oslo,
NO-0316 Oslo, Norway
Department of Biological Sciences, SUNY Buffalo,
Buffalo, NY 14260, USA
Email: victor.albert@nhm.uio.no

Claude dePamphilis
Biology Department, Penn State University,
University Park, PA 16802, USA
Email: cwd3@psu.edu

David Sankoff*
Department of Mathematics and Statistics, University of Ottawa,
Ottawa, Ontario K1N 6N5, Canada
**Email: sankoff@uottawa.ca*

We improve on guided genome halving algorithms so that several thousand gene sets, each containing two paralogs in the descendant T of the doubling event and their single ortholog from an undoubled reference genome R, can be analyzed to reconstruct the ancestor A of T at the time of doubling. At the same time, large numbers of defective gene sets, either missing one paralog from T or missing their ortholog in R, may be incorporated into the analysis in a consistent way. We apply this genomic rearrangement distance-based approach to the recently sequenced poplar (*Populus trichocarpa*) and grapevine (*Vitis vinifera*) genomes, as T and R respectively.

1. INTRODUCTION

Following an episode of whole genome doubling, intra- and interchromosomal rearrangement processes over evolutionary time redistribute segments both large and small across the genome. The present-day genome can be largely decomposed into a set of duplicated DNA segments dispersed among the chromosomes, with all the duplicate pairs exhibiting a similar degree a sequence divergence. A linear-time "genome halving" algorithm, based only

*Corresponding author.

on the chromosomal segment order, can find an ancestral genome that minimizes the genomic distance to the present-day genome[1, 2]. Unfortunately, the output of the combinatorial optimization method does not really suffice as a solution to the reconstruction problem, since there may be a large number of very different, equally optimal solutions. Our guided genome halving (GGH) strategy to oversome this non-uniqueness problem is to guide the reconstruction of the ancestor by one or more reference, or outgroup, genomes. This strategy does not imply sacrificing optimality of the halving solution.

The flowering plants are well-known for numerous historical events of genome doubling[3]. The recently sequenced poplar genome (*Populus trichocarpa*)[4], which shows very clear evidence of recent genome duplication, and the grapevine genome (*Vitis vinifera*)[5, 6], whose ancestor diverged before the aforementioned duplication, provide a pair of analytical incentives to the GGH strategy. On the one hand, the poplar data has an order of magnitude more duplicated elements than has previously been analyzed, straining computational resources. On the other hand, the richness of the data allows us to assess the effects on ancestral genome reconstruction of the apparently massive loss of duplicate genes, as suggested by the modest proportion of paralogous pairs we can detect, as the poplar genome discarded most of its duplications.

This paper thus contributes two advances on the methodological level: first, the scaling up, by more than an order of magnitude, of the amount of data amenable to our analysis, and second, the incorporation of data from gene duplicate pairs that have lost one member, making use of chromosomal context in both the genome that can be traced to the doubling event and in the outgroup.

1.1. Background

Algorithms for guided genome halving (GGH), or reconstruction of the pre-doubling genome with the help of an outgroup, were first used for the ancestral doubled genome of the maize (*Zea mays*) genome, with the rice (*Oryza sativa*) and sorghum (*Sorghum bicolor*) genomes as outgroups[7]. We generated all the 1.5×10^6 solutions to the genome halving problem for the maize genome, and then identified the subset, containing only a handful of relatively similar solutions that have a minimum rearrangement distance with the rice (or sorghum) genome.

This approach was feasible with the small number (34) of doubled blocks identified in maize that were also present in one copy in each outgroup, but in a subsequent analysis[8], when we attempted to reconstruct the ancient doubled yeast genome from which *Saccharomyces cerevisiae* is descended, guided simultaneously by both of the undoubled outgroup genomes *Ashbya gossypii* and *Kluyveromyces waltii*, the number of doubled genes we could use as evidence was an order of magnitude greater than the number of blocks in the cereals data, and the number of solutions to the halving problem astronomical. It was no longer feasible to exhaustively search the halving solutions to find those that are closest to the outgroups. Instead we took a random sample of several thousand solutions in the hope that the best one might be optimal, or close to it. It was not clear, however, how large the sample should be, or how to validate the results, since the local optima found in that study remained fairly far apart, as measured by genomic rearrangement distance.

In our current use of GGH, on yeast[9] and on the flowering plants studied in the present article, we seek to replace the brute force approach of generating all (or a random sample of) halving solutions first, i.e., before taking into consideration the outgroup genome. Instead, we inject all pertinent information derivable from the outgroup into the halving algorithm, influencing hitherto arbitrary choices in that algorithm so that the halving solution is guided towards the outgroup.

1.2. Outline

In the next section, we sketch the necessary background about genomic rearrangement distance and the genome halving and GGH algorithms. In Section 3, we describe the sources for our data and how we processed them to obtain the gene sets for the GGH analysis. In Section 4 we present the GGH algorithm incorporating both full and defective gene sets. We apply this method to the full gene sets in combination with one or both of two defective gene sets from *Populus* and *Vitis* in Section 5. We present the reconstructed undoubled *Populus* ancestor based on over 6000 gene sets and evaluate the evolutionary signal versus noise (a) in the ancestor-*Populus* and ancestor-*Vitis* comparisons, (b) in the full and defective gene sets, and (c) in genes with two or three common adjacencies in the data and those with weaker positional evidence.

2. FORMAL PRELIMINARIES AND PREVIOUS WORK

2.1. Genomes, rearrangement operations and genomic distance

A genome G is represented by a set of strings (called *chromosomes*) of form $\{g_{11}\cdots g_{1n_1},...,g_{\chi 1}\cdots g_{\chi n_\chi}\}$, where $n = n_1+\cdots+n_\chi$ and $\{|g_{..}|\} = \{1,\cdots,n\}$; i.e., each integer $i \in \{1,\cdots,n\}$ appears exactly once in the genome and may have either positive or negative polarity. The biologically-motivated operations of reversal or inversion, reciprocal translocation, chromosome fission or fusion, and transposition, can all be represented by an operation (called double-cut and join, or DCJ) of cutting the genome twice, each time between two elements on one of the chromosomes and rejoining the four resulting cut ends differently[10, 11]. Whether the two cuts are on the same chromosome or not, and how the endpoints are rejoined, determine which rearrangement operation pertains.

The genome rearrangement distance $d(G,H)$ is defined to be the minimum number of DCJ operations required to convert one of the genomes, G, into the other, H.

Rearrangement algorithms[12, 13, 10] can be formulated in terms of the bi-coloured "breakpoint graph", where each end (either $5'$ or $3'$) of a gene in genome G is represented by a vertex joined by a black edge to the vertex for adjoining end of the adjacent gene, and these same ends, represented by the same $2n$ vertices in the graph, are joined by gray edges determined by the adjacencies in genome H. In addition, each vertex representing a first or last term of some chromosome in G or in H is connected by a edge of the appropriate colour to an individual "cap" vertex, and there are specific rules for adding caps to the genome with fewer chromosomes and for joining the caps among themselves. if G has χ chromosomes and H has no more than χ, there are $2n+4\chi$ vertices in all. The breakpoint graphs necessarily consist of disjoint alternating colour cycles, and it can be shown that, in the DCJ formulation, $d(G,H) = n+\chi-c$, where c is the number of cycles in the breakpoint graph. Calculating the distance can be done in time linear in n.

2.2. Genome halving

Let T be a genome consisting of ψ chromosomes and $2n$ genes $a_1^{(1)}\cdots,a_n^{(1)};a_1^{(2)},\cdots,a_n^{(2)}$, dispersed in any order on the chromosomes. For each i, we call $a_i^{(1)}$ and $a_i^{(2)}$ "duplicates", but there is no particular property distinguishing all elements of the set of $a_i^{(1)}$ in common from all those in the set of $a_i^{(2)}$. A potential "doubled ancestor" of T is written $A' \oplus A''$, and consists of 2χ chromosomes, where some half (χ) of the chromosomes, symbolized by the A', contains exactly one of $a_i^{(1)}$ or $a_i^{(2)}$ for each $i = 1,\cdots,n$. The remaining χ chromosomes, symbolized by the A'', are each identical to one in the first half, in that where $a_i^{(1)}$ appears on a chromosome in the A', $a_i^{(2)}$ appears on the corresponding chromosome in A'', and where $a_i^{(2)}$ appears in A', $a_i^{(1)}$ appears in A''. We define A to be either of the two halves of $A' \oplus A''$, where the superscript (1) or (2) is suppressed from each $a_i^{(1)}$ or $a_i^{(2)}$. *The genome halving problem for T is to find an A for which some $d(A' \oplus A'', T)$ is minimal.*

In the rearrangement distance algorithm, construction of the breakpoint graph is an easy step. The genome halving algorithms [2] also make use of the breakpoint graph, but the problem here is the more difficult one of building the breakpoint graph where one of the genomes (the doubled ancestor $A' \oplus A''$) is unknown. This is done by segregating the vertices of the graph in a natural way into subsets, such that all the vertices of each cycles must fall within a single subset, and then constructing these cycles in an optimal way within each subset so that the black edges correspond to the structure of the known genome T and the gray edges define the adjacencies of $A' \oplus A''$.

As a first step each gene a in a doubled descendant is replaced by a pair of vertices (a_t, a_h) or (a_h, a_t) depending if the DNA is read from left to right or right to left. The duplicate of gene $a = (a_t, a_h)$ is written $\bar{a} = (\bar{a_t}, \bar{a_h})$.

Following this, for each pair of neighbouring genes, say (a_t, a_h) and (b_h, b_t), the two adjacent vertices a_h and b_h are linked by a black edge, denoted $\{a_h, b_h\}$ in the notation of Ref. 11. For a vertex at the end of a chromosome, say b_t, it generates a virtual edge of form $\{b_t, \text{end}\}$. Note that the use of "end" instead of "cap" reflects a somewhat different bookkeeping for the beginnings and ends of chromosome in the halving algorithm compared to the distance algorithm in Section 2.1.

The edges thus constructed are then partitioned into *natural graphs* according to the following principle: If an edge $\{x, y\}$ belongs to a natural graph, then so does some edge of form $\{\bar{x}, z\}$ and some edge of form $\{\bar{y}, w\}$. If a natural graph has an even number of edges, it can be shown that in all optimal ancestral doubled genomes, the edges coloured gray, say, representing adjacent vertices in the ancestor, and incident to one of the vertices in this natural graph, necessarily have as their other endpoint another vertex within the same natural graph.

For all other natural graphs, there are one or more ways of grouping them pairwise into *supernatural graphs* so that an optimal doubled ancestor exists such that the edges coloured gray incident to any of the vertices in a supernatural graph have as their other endpoint another vertex within the same supernatural graph. Thus the supernatural graph may be completed one at a time.

An important detail in this construction is that before a gray edge is added during the completion of a supernatural graph, it must be checked to see that it would not inadvertently result in a circular chromosome. Key to the linear worst-case complexity of the halving algorithm is that this check may be made in constant time.

Along with the multiplicity of solutions caused by different possible constructions of supernatural graphs, within such graphs and within the natural graphs, there may be many ways of drawing the gray edges. Without repeating here the lengthy details of the halving algorithm, it suffices to note that these alternate ways can be generated by choosing one of the vertices within each supernatural graph as a starting point.

2.3. Genome halving with outgroups

Let T be a genome consisting of ψ chromosomes and $2n$ genes $a_1^{(1)} \cdots , a_n^{(1)}; a_1^{(2)}, \cdots , a_n^{(2)}$, dispersed in any order on the chromosomes, where for each i, genes $a_i^{(1)}$ and $a_i^{(2)}$ are duplicates. Any genome R is a reference or outgroup genome for T if it contains the n genes $a_1, \cdots , a_n$.

Let R be a reference genome for T. The GGH problem with one outgroup is to find a potential ancestral genome A such that some $d(R, A) + d(A' \oplus A'', T)$ is minimal. In practice, A is either one of the solutions to the unconstrained halving problem, or it is close to such a solution, so little is lost in restricting our search to the set of solutions of the genome halving problem for T.

One strategy, suitable for small data sets, as in Ref. 7, is to generate the entire set $\mathbf{S}$ of genome halving solutions of T, then to evaluate each $A \in \mathbf{S}$ to find the one that minimizes $d(R, A)$.

When $\mathbf{S}$ is so large that it is not feasible to generate all of $\mathbf{S}$ in order to find the best A, we may resort to sampling $\mathbf{S}$, as in Ref. 8. In defining the gray edges in the supernatural graphs of Section 2.2, we generally have several choices at some of the steps. By randomizing this choice, we are effectively choosing a random sample of $X \in \mathbf{S}$.

3. THE POPULUS-VITIS COMPARISON

Annotations for the *Populus* and *Vitis* genomes were obtained from databases maintained by the U.S. Department of Energy's Joint Genome Institute[4] and the French National Sequencing Center, Genoscope[6], respectively. An all-by-all BLASTP search was run on a data set including all *Populus* and *Vitis* protein coding genes, and orthoMCL[14] was used to construct 2104 full and 4040 defective gene sets, in the first case containing two poplar paralogs (genome T) and one grape ortholog (genome R), and in the second case missing a copy from either T or R. The chromosomal location and orientation of these paralogs and orthologs was used to construct our database of gene orders for these genomes, and the input to the GGH algorithm.

4. THE GGH ALGORITHM

The key idea in our improvement over brute force algorithms is to incorporate information from R during the halving process. It is important to take advantage of the common structure in T and R as early as possible, before it can be destroyed in the course of construction. To this end, we drop the practice of completing all the gray edges in one supernatural graph before starting another. We simply look for elements of common structure and add gray edges accordingly, always making sure that no circular chromosomes are inadvertently created.

Missing homologs The halving algorithm requires full gene sets at several steps in reconstructing the ancestor, so we algorithmically restore the missing homologs to appropriate positions in T and R at the outset.

Paths We define a path to be any connected fragment of a breakpoint graph, namely any connected fragment of a cycle. We represent each path by an unordered pair $(u, v) = (v, u)$ consisting of its current endpoints, though we keep track of all its vertices and edges. Initially, each black edge in T is a path, and each black edge in R is a path.

Pathgroups A pathgroup Γ is an ordered triple of paths, two in T and one in R, where one endpoint of one of the paths in T is the duplicate of one endpoint of the other path in T and both are orthologous to one of the endpoints of the path in R. The other endpoints may be duplicates or orthologs to each other, or not.

4.1. The algorithms

In adding pairs of gray edges to connect duplicate pairs of terms in the breakpoint graph of T versus $A' \oplus A''$, (which is being constructed), our approach is basically greedy, but with a sophisticated look-ahead. We can distinguish five different levels of desirability, or priority, among potential gray edges, i.e., potential adjacencies in the ancestor.

Recall that in constructing the ancestor A to be close to the outgroup R, such that $A' \oplus A''$ is simultaneously close to T, we must create as many cycles as possible in the breakpoint graphs between A and R and in the breakpoint graph of $A' \oplus A''$ versus T.

(1) Adding two gray edges would create two cycles in the breakpoint graph defined by T and $A' \oplus A''$, by closing two paths. When this possibility exists, it must be realized, since it is an obligatory choice in any genome halving algorithm. It may or may not also create cycles in the breakpoint graph comparison of X with the outgroup, but this does not affect its priority.

(2) Adding two gray edges would create two cycles, one for T and one for the outgroup.

(3) Adding two gray edges would create a cycle in the T versus $A' \oplus A''$ comparison, but none for the outgroup. It would, however, create a higher priority pathgroup.

(4) Adding two gray edges would create a cycle in the T versus $A' \oplus A''$ comparison, but none for the outgroup, nor would it create any higher priority pathgroup.

(5) Each remaining path terminates in duplicate terms, which cannot be connected to form a cycle, since in $A' \oplus A''$ these must be on different (and identical) chromosomes. In supernatural graphs containing such paths, there is always another path and adding two gray edges between the endpoints of the two paths can create a cycle.

In not completing each supernatural graph before moving on to another, we lose the advantage in Ref. 2 of a constant time check against creating circular chromosomes. The worst case becomes a linear time check. This is a small liability, because the worst case scenario is seldom realized, the check almost always requiring only one or two steps.

Algorithm GGH:

Guided Genome Halving with Full and Defective Gene Sets

```
Input. Two genomes: duplication descendant $T'$, outgroup genome $R'$, where each gene is has three
    homologs (full set) or two homologs (defective set), in the patterns TTR, TT or TR.
Output. Augmented genomes $T$, and $R$, where all gene sets are full, and
    Genome $A$, a halving solution of $T$, minimizing $d(A' \oplus A'', T) + d(A, R)$.
insertMH
Initialize paths (black edges) in $T$ and $R$.
Construct supernatural graphs.
Construct two pathgroups for each gene $g$ in $R$, one based on $g_t$, the other on $g_h$.
If number of chromosomes in $T$ is odd,
    add pathgroup with two paths of form $(\mathrm{end}, \mathrm{end})$.
While there remains at least one pathgroup
    For each pathgroup $((x, y), (\bar{x}, z), (x, m))$
    classify it by case and priority, and find a pathgroup $\Gamma$ that has the highest priority. To choose among
    Priority 2 pathgroups, find one that maximizes the number of “real” black edges, i.e., edges in $T'$ and $R'$,
    not just edges created by insertMH. Similarly for Priority 3 pathgoups.
    Case 1: $\bar{x} \neq y$, and adding $xy$ and $\bar{x}\bar{y}$ would not create a circular chromosome.
      Priority 1: $z = \bar{y}$.
      Priority 2: $y = m$.
      Priority 3: adding $xy$ and $\bar{x}\bar{y}$ would create a pathgroup with priority 2.
      Priority 4: None of 1, 2 or 3.
    Case 2: $\bar{x} \neq y$, and adding $x\bar{z}$ and $\bar{x}z$ would not create a circular chromosome.
      Priority 2: $z = m$.
      Priority 3: adding $x\bar{z}$ and $\bar{x}z$ would create a pathgroup with priority 2.
      Priority 4: Neither of 2 or 3.
    Case 3:$\bar{x} = y$.
      Priority 5:
    If $\Gamma$ is Case 1, addGrayEdge($xy, \bar{x}\bar{y}$).
    If $\Gamma$ is Case 2, addGrayEdge($x\bar{z}, \bar{x}z$).
    If $\Gamma$ is Case 3, find some
      $W = ((w, \bar{w}), (\bar{w}, w), (w, s))$ in the same supernatural graph and addGrayEdge($xw, \bar{x}\bar{w}$).
```

Algorithm: addGrayEdge$(rt, \bar{r}\bar{t})$

Add gray edges $rt, \bar{r}\bar{t}$ to partially completed genome X” $\oplus X''$.

Add gray edge rt to partially completed genome X.

Update paths in pathgroups that are affected by the new gray edges.

Remove pathgroups that start with r and t.

Algorithm: insertMH:

Insert Missing Homologs in Chromosomes

Input. Two genomes: duplication descendant T', outgroup R', where each gene is has two or three homologs, in the patterns TTR, TT, TR.

Output. Augmented genomes T and R containing exactly three homologs for each gene, in the pattern TTR, maximizing the number of common edges of form $\{a_1, b_1\}$, $\{a_2, b_2\}$ in T and $\{a, b\}$ in R.
(Or $\{a_1, b_2\}$, $\{a_2, b_1\}$ in T and $\{a, b\}$ in R.)

While there are genes that have only two copies, **count edgeDiff** for each such, which simultaneously finds the BestPosition.

Insert the gene with the minimum edgeDiff value into the BestPosition of this gene.

Algorithm: count edgeDiff

If a gene g just has one copy (g_1) in T' and one copy (g) in R', then we must insert another copy (g_2) into T'.

If a gene g just has two copies (g_1, g_2) in T', then we must insert g into R'.

(The details are omitted here. This is essentially a greedy heuristic to add adjacencies reflecting, as if possible, adjacencies already existing in R' and T'.)

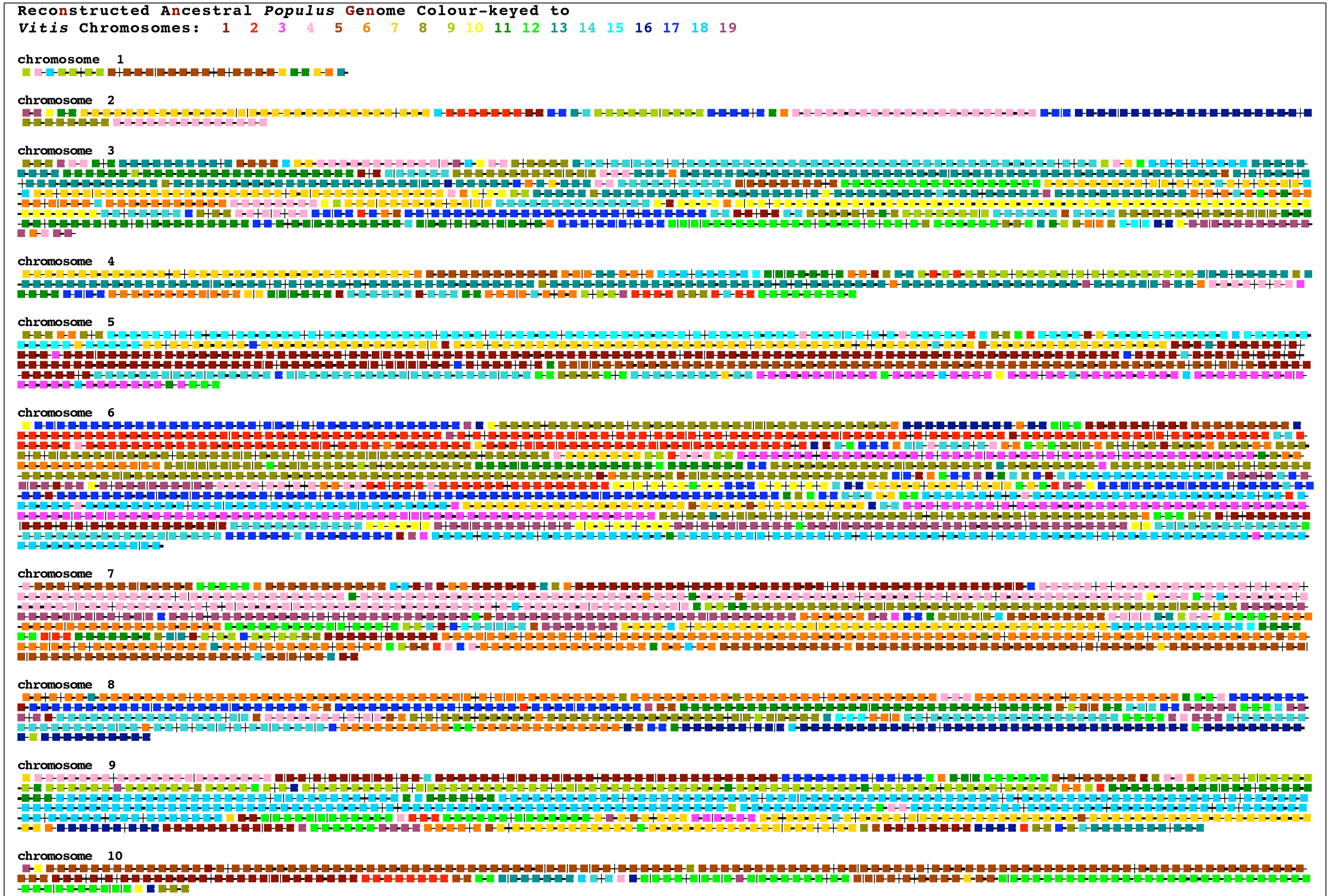

Fig. 1. Ten chromosomes (wrapped) of ancestral poplar genome reconstructed by GGH algorithm from 6144 full and defective gene sets. Only genes with grape orthologs are indicated. Adjacencies present three times, i.e., twice in poplar and once in grape are indicated by ≡, those present twice by = and those present once by −. Intrachromosomal breakpoints within segments indicated by |.

5. RESULTS AND DISCUSSION

Our data consisted of 6144 gene sets, of which only 2104 were full sets. There were only 836 defective sets by virtue of a missing ortholog in V, while 3204 genes lacked one paralog in T.

Table 1. Comparisons of the reconstructed immediate pre–doubling ancestor A with the *Vitis* genome and of the immediate doubled ancestor $A \oplus A$ with *Populus*. PPV: full gene sets, PP: defective, missing grape ortholog, PV: defective, missing one poplar paralog. Projected: genes not in PPV ancestor deleted from solution A, d: genomic distance, b:,number of breakpoints, $r = 2d/b$: the re-use statistic.

	genes	$d(A, Vitis)$			$d(A{\oplus}A, Populus)$		
data sets	in A	d	b	r	d	b	r
PPV	2104	638	751	1.70	454	690	1.32
PPV,PP	2940	649	757	1.71	737	1090	1.35
projected	2104	649	757	1.71	581	823	1.41
PPV,PV	5308	1180	1331	1.77	1083	1457	1.49
projected	2104	663	758	1.75	670	833	1.61
PPV,PP, PV	6144	1208	1363	1.77	1337	1812	1.48
projected	2104	664	757	1.75	750	926	1.62
without singletons							
PPV	2020	560	661	1.69	346	541	1.28
PPV,PP	2729	594	690	1.72	453	714	1.27
projected	2006	571	664	1.72	416	628	1.32
PPV,PV	4203	573	686	1.67	751	1031	1.46
projected	1955	489	580	1.69	490	644	1.52
PPV,PP, PV	4710	675	797	1.69	856	1211	1.41
projected	1986	528	622	1.70	558	744	1.50

Table 1 shows the results of the analysis on the full gene sets only, on combinations of the full sets with one kind of defective sets, and all three sets. For each case we study not only the reconstructed ancestor but also a "projected" version where genes from the defective sets are simply erased, in order to assess the changes in gene order due to the defective gene sets. Whereas the distance between each T and its reconstructed ancestor A is given by GGH, the distance between projected ancestor and T required a heuristic, not detailed here, for attributing each paralog in T to one of the two copies of the ancestral genome.

Figure 1 depicts the result of analyzing all the 6144 gene sets with GGH, although the 836 genes with no grape ortholog are not visible. The large number of singleton genes disrupting otherwise homogeneous synteny blocks suggests that "noise" due to uncertainties inherent in homology identification and especially orthology identification may be artifactually inflating genomic distance d and the number of breakpoints b. Since the rigorous noise elimination techniques of Refs. 15 and 16 have not yet been extended in the context of genome doubling, we simply identified singletons as gene sets lacking two real (i.e., not inferred from **insertMH**) common adjacencies out of six possible in the original genomes, and ran all the analyses again without these genes.

In each case, we counted the breakpoints and calculated the appropriate genomic distance d, i.e., from the doubled ancestor to *Populus* and from the undoubled version of the same ancestor to *Vitis*. This enabled us to calculate the "breakpoint re-use" statistic $r = 2d/b$, which is a measure of how much signal about conserved order (among segments, not within segments) remains in the comparison of two genomes after a period of evolutionary rearrangements. When $r = 1$, we can have high confidence in the rearrangement distance and history. When r approaches two, the segment order in the two genomes being compared are essentially random with respect to each other, i.e., calculating r for random genomes gives a value approaching 2[a]. In Table 1, we see both from changes in d and changes in r that

- most of the signal contained in the order among conserved chromosomal segments has been lost between the ancestor and *Vitis*, but is retained to a great degree between the ancestor and *Populus*, probably reflecting the difference in divergence time but also possible biases towards T in the GGH algorithm,
- the addition of the defective PV gene sets degrades the analysis, more than the addition of PP sets, though this may due to the four times greater number of gene sets in the former,
- the elimination of singletons improves all the analyses, but where PV is present, this comes about largely by discarding most of the sets, which turn out to be singletons.

With the application of our method to the more than

[a] If breakpoints are frequently re-used during evolution, then r will also be close to 2; unfortunately there is no internal way of testing the breakpoint re-use hypothesis against the null hypothesis of complete loss of signal about segment order[17].

6000 gene sets, we have shown that any realistic case of genome doubling should be amenable, even if all the gene paralogs remain in the sequenced descendant. The analysis with 6144 gene sets required almost 48 hours on a MacBook, but this was anomalously large, since those with 4000 or 5000 required less than 5 hours and those with 2000 about 1 hour. Much of the running time is due to the check on the number of real edges in a pathgroup to choose among Priority 2 or among Priority 3 options. This could be reduced by optimizing data structures in our software.

The inclusion of defective PV gene sets would appear to add little more than noise to the analysis, but the PP sets would seem to add significant information, especially to the ancestor-*Populus* comparison.

The elimination of singletons proves to be a meaningful way of drastically decreasing the number of segments (as measured by b) and the genomic distance to credible levels, though this still does not result in a detectible signal in the ancestor-*Vitis* comparison. The recently sequenced and asembled *Carica papaya* genome, which is phylogenetically more closely related to *Populus*, but like *Vitis* diverged before the *Populus* doubling event, should be better able play the outgroup role in our analysis, once it is published and we have been able to identify orthologs.

Acknowledgments

Research supported in part by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC). DS holds the Canada Research Chair in Mathematical Genomics.

References

1. El-Mabrouk N, Bryant D, Sankoff D. Reconstructing the pre-doubling genome. In: Istrail S, Pevzner P, Waterman M (eds.), Third Annual International Conference on Computational Molecular Biology (RECOMB 99). ACM Press, New York. 1999: 154–163.
2. El-Mabrouk N, Sankoff D. The reconstruction of doubled genomes. *SIAM Journal on Computing* 2003; **32**: 754–792.
3. Cui L, Wall PK, Leebens-Mack JH, Lindsay BG, Soltis DE, Doyle JJ, Soltis PS, Carlson JE, Arumuganathan K, Barakat A, Albert VA, Ma H, dePamphilis CW. Widespread genome duplications throughout the history of flowering plants. *Genome Research* 2006; **16**: 738–749.
4. Tuskan GA, Difazio S, Jansson S, Bohlmann J, Grigoriev I, Hellsten U, Putnam N, Ralph S, Rombauts S, Salamov A, Schein J, Sterck L, Aerts A, Bhalerao RR, Bhalerao RP, Blaudez D, Boerjan W, Brun A, Brunner A, Busov V, Campbell M, Carlson J, Chalot M, Chapman J, Chen GL, Cooper D, Coutinho PM, Couturier J, Covert S, Cronk Q, Cunningham R, Davis J, Degroeve S, Djardin A, Depamphilis C, Detter J, Dirks B, Dubchak I, Duplessis S, Ehlting J, Ellis B, Gendler K, Goodstein D, Gribskov M, Grimwood J, Groover A, Gunter L, Hamberger B, Heinze B, Helariutta Y, Henrissat B, Holligan D, Holt R, Huang W, Islam-Faridi N, Jones S, Jones-Rhoades M, Jorgensen R, Joshi C, Kangasjrvi J, Karlsson J, Kelleher C, Kirkpatrick R, Kirst M, Kohler A, Kalluri U, Larimer F, Leebens-Mack J, Lepl JC, Locascio P, Lou Y, Lucas S, Martin F, Montanini B, Napoli C, Nelson DR, Nelson C, Nieminen K, Nilsson O, Pereda V, Peter G, Philippe R, Pilate G, Poliakov A, Razumovskaya J, Richardson P, Rinaldi C, Ritland K, Rouz P, Ryaboy D, Schmutz J, Schrader J, Segerman B, Shin H, Siddiqui A, Sterky F, Terry A, Tsai CJ, Uberbacher E, Unneberg P, Vahala J, Wall K, Wessler S, Yang G, Yin T, Douglas C, Marra M, Sandberg G, Van de Peer Y, Rokhsar D. The genome of black cottonwood, Populus trichocarpa (Torr. & Gray). *Science* 2006; **313**: 1596–1604. http://genome.jgi-psf.org/Poptr1/Poptr1.download .html
5. Velasco R, Zharkikh A, Troggio M, Cartwright DA, Cestaro A, Pruss D, Pindo M, Fitzgerald LM, Vezzulli S, Reid J, Malacarne G, Iliev D, Coppola G, Wardell B, Micheletti D, Macalma T, Facci M, Mitchell JT, Perazzolli M, Eldredge G, Gatto P, Oyzerski R, Moretto M, Gutin N, Stefanini M, Chen Y, Segala C, Davenport C, Dematt L, Mraz A, Battilana J, Stormo K, Costa F, Tao Q, Si-Ammour A, Harkins T, Lackey A, Perbost C, Taillon B, Stella A, Solovyev V, Fawcett JA, Sterck L, Vandepoele K, Grando SM, Toppo S, Moser C, Lanchbury J, Bogden R, Skolnick M, Sgaramella V, Bhatnagar SK, Fontana P, Gutin A, Van de Peer Y, Salamini F, Viola R. A high quality draft consensus sequence of the genome of a heterozygous grapevine variety. *PLoS ONE* 2007; **2**: e1326.
6. Jaillon O, Aury JM, Noel B, Policriti A, Clepet C, Casagrande A, Choisne N, Aubourg S, Vitulo N, Jubin C, Vezzi A, Legeai F, Hugueney P, Dasilva C, Horner D, Mica E, Jublot D, Poulain J, Bruyre C, Billault A, Segurens B, Gouyvenoux M, Ugarte E, Cattonaro F, Anthouard V, Vico V, Del Fabbro C, Alaux M, Di Gaspero G, Dumas V, Felice N, Paillard S, Juman I, Moroldo M, Scalabrin S, Canaguier A, Le Clainche I, Malacrida G, Durand E, Pesole G,

Laucou V, Chatelet P, Merdinoglu D, Delledonne M, Pezzotti M, Lecharny A, Scarpelli C, Artiguenave F, P ME, Valle G, Morgante M, Caboche M, Adam-Blondon AF, Weissenbach J, Qutier F, Wincker P; French-Italian Public Consortium for Grapevine Genome Characterization. The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla. *Nature* 2007; **449**: 463–467. http://www.genoscope.cns.fr/externe/English/Projets/Projet_ML/data/annotation/

7. Zheng C, Zhu Q, Sankoff D. Genome halving with an outgroup. *Evolutionary Bioinformatics* 2006; **2**: 319–326.
8. Sankoff D, Zheng C, Zhu Q. Polyploids, genome halving and phylogeny. *Bioinformatics* 2007; **23**: i433–i439.
9. Zheng C, Zhu Q, Adam Z, Sankoff D. Guided genome halving: hardness, heuristics and the history of the Hemiascomycetes. *Bioinformatics* 2008; **24**.
10. Yancopoulos S., Attie O, Friedberg R. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics* 2005; **21**: 3340–3346
11. Bergeron A, Mixtacki J, Stoye J. A unifying view of genome rearrangements. In: Bücher P, Moret BME (eds.), Workshop on Algorithms in Bioinformatics (WABI 2006). Lecture Notes in Computer Science **4175**, 2006:163–173.
12. Bafna V, Pevzner P. Genome rearrangements and sorting by reversals. *SIAM Journal of Computing* 1996; **25**: 272–289.
13. Tesler G. Efficient algorithms for multichromosomal genome rearrangements. *Journal of Computer and System Sciences* 2002; **65**: 587–609.
14. Li L, Stoeckert CJ Jr, Roos DS. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Research* 2003; **13**: 2178–2189.
15. Zheng C, Zhu Q, Sankoff D. Removing noise and ambiguities from comparative maps in rearrangement analysis. *Transactions on Computational Biology and Bioinformatics* 2007; **4**: 515–522.
16. Choi V, Zheng C, Zhu Q, Sankoff D. Algorithms for the extraction of synteny blocks from comparative maps. In: Giancarlo R, Hannenhalli S. (eds.), Workshop on Algorithms in Bioinformatics (WABI 2007). Lecture Notes in Bioinformatics **4645**, 2007: 277–288.
17. Sankoff D. The signal in the genomes. *PLoS Computational Biology* 2006; **2**: e35.