Generalized Gene Adjacencies, Graph Bandwidth and Clusters in Yeast Evolution

Qian Zhu[1], Zaky Adam[1], Vicky Choi[2], and David Sankoff[1]

[1] Department of Biochemistry, School of Information Technology and Engineering,
and Department of Mathematics and Statistics,
University of Ottawa, Ottawa, Canada K1N 6N5
{qzhu012,zadam008,sankoff}@uottawa.ca

[2] Department of Computer Science, Virginia Tech., Blacksburg, VA 24061
vchoi@cs.vt.edu

Abstract. We present a parametrized definition of gene clusters that allows us to control the emphasis placed on conserved order within a cluster. Though motivated by biological rather than mathematical considerations, this parameter turns out to be closely related to the maximum bandwidth parameter of a graph. Our focus will be on how this parameter affects the characteristics of clusters: how numerous they are, how large they are, how rearranged they are and to what extent they are preserved from ancestor to descendant in a phylogenetic tree. We infer the latter property by dynamic programming optimization of the presence of individual edges at the ancestral nodes of the phylogeny. We apply our analysis to a set of genomes drawn from the Yeast Gene Order Browser.

1 Introduction

The definition of synteny blocks, gene clusters or similar constructs from the comparison of two or more genomes entails a trade-off of great consequence: if we place emphasis on identical content and order of the genes, segments or markers in a block or cluster, only relatively small regions of the genome will satisfy this restrictive condition, giving rise to a plethora of tiny blocks while missing large regions common to the genomes. On the other hand, by allowing unrestricted scrambling of genes within blocks (e.g., max-gap [1] or "gene teams" [7]), we forgo accounting for local genome rearrangement, missing an important aspect of evolutionary history, or we relinquish the possibility of pinpointing extensive local conservation, where this exists.

In this paper, we present a parametrized definition of gene clusters that allows us to control the emphasis placed on conserved order within a cluster. Though motivated by biological rather than mathematical considerations, this parameter turns out to be closely related to the maximum bandwidth parameter of a graph. Our focus will be on how this parameter affects the characteristics of clusters: how numerous they are, how large they are, how rearranged they are and to

I. Măndoiu, R. Sunderraman, and A. Zelikovsky (Eds.): ISBRA 2008, LNBI 4983, pp. 134–145, 2008.

what extent they are preserved from ancestor to descendant in a phylogenetic tree. We infer the latter property by dynamic programming optimization of the presence of individual edges in a generalized adjacency graph abstractly representing chromosomal gene order. We apply our analysis to a set of genomes drawn from the Yeast Gene Order Browser (YGOB) [3]. Among the results, we find strong evidence for setting a certain fixed value to the cluster parameter. We also find that we can recover almost all the clusters that can be found without order constraints, i.e., by the max-gap criterion, indicating that local order conservation is a lot greater than that unconstrained definition would suggest.

2 Definitions

Our characterization of gene clusters is made up of a general part that identifies clusters of vertices common to two graphs, and a specific part where a graph is determined by the proximity of genes on the chromosomes of a genome. This is illustrated in Figure 1.

Definition 1. *Let $G_S = (V_S, E_S)$ and $G_T = (V_T, E_T)$ be two graphs with a non-null set of vertices in common $V = V_S \cap V_T$. We say a subset of $C \subseteq V$ is an ST-***cluster** *if it consists of the vertices of a maximal connected subgraph of $G_{ST} = (V, E_S \cap E_T)$.*

Definition 2. *For the purposes of genome comparison, we may consider V_X to be the set of genes in the genome X. For genes g and h in V_X on the same chromosome in X, let $gh \in E_X$ if the number of genes intervening between g and h in X is less than θ, where $\theta \geq 1$ is a fixed* **neighbourhood parameter**.

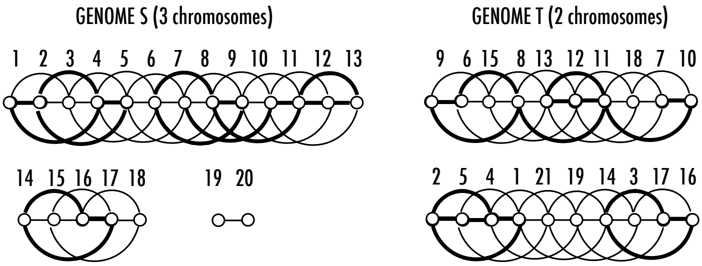
These definitions of edge sets and ST-clusters decompose the genes in the two genomes into identical sets of disjoint clusters of size greater or equal to 2, and possibly different sets of singletons belonging to no cluster, either because they are in V, but not in $E_S \cap E_T$, or because they are in $(V_S \cup V_T \setminus V)$. For simplicity, we do not attempt to deal with duplicate genes in this paper. When $\theta = 1$, a cluster has exactly the same gene content and order (or reversed order) in both genomes. When $\theta = \infty$, the definition returns simply all the synteny sets, namely the sets of genes in common between two chromosomes, one in each genome.

Let Π be the set of all orderings of V. Recall that the **bandwidth** of a graph $G(V, E)$ is defined to be

$$B(G) = \min_{p \in \Pi} \max_{uv \in E} |p(u) - p(v)|. \tag{1}$$

In a genome S each chromosome χ determines a physical order among the genes it contains.

Proposition 1. *Bandwidth $B(G_S) = \theta$, as long there are at least $2\theta + 1$ genes on some chromosome χ in genome S.*



ST clusters

θ= 2 : {2,4,5}, {6,8}, {11, 12 13}, {16, 17}
θ= 3 : {1,2,4,5}, {6,7,8,9,10, 11, 12 13}, {14, 16, 17}
θ= 4 : {1,2,3,4,5}, {6,7,8,9,10, 11, 12 13}, {14, 16, 17}

Fig. 1. Graphs constructed from two genomes using parameter $\theta = 3$. Thick edges determine clusters. ST-clusters listed for $\theta = 2$ and $\theta = 4$ as well.

Proof: By Definition 2, the vertex v corresponding to the gene at position $\theta+1$ on chromosome χ is connected to 2θ other vertices. The most remote of these are at positions 1 and $2\theta+1$. Similarly, for a vertex u at any other position on χ, we can show that the most remote gene connected to u is no farther away than θ. Thus, for the order $p(\cdot)$ on the vertices defined by the original gene order, $\max|p(u)-p(v)| = \theta$. Hence, $B(G_S) \leq \theta$.

For any other order $p(\cdot)$, consider the 2θ vertices connected to vertex v. For one such vertex w, $|p(v)-p(w)| \geq \theta$, since we cannot fit 2θ vertices connected to v into an interval of size $< 2\theta+1$, also containing v.

Since the upper and lower bounds coincide, the proposition follows. □

Proposition 2

$$B(G_{ST}) = \max_{C\in\mathcal{C}} B(C), \tag{2}$$

where $\mathcal{C}$ is the set of connected components of G_{ST}.

Proof: Since E_{ST} is the union of the edges in all the C,

$$\max_{uv\in E_{ST}} |p(u)-p(v)| = \max_{C\in\mathcal{C}} \max_{uv\in E_C} |\bar{p}(u)-\bar{p}(v)|, \tag{3}$$

where $\bar{p}(\cdot)$ is the order induced on the vertices in C by the order $p(\cdot)$ on E_{ST}. But any set of vertex orders on all the individual C can be jointly extended to an order on V_{ST}. □

We compare the definition of an ST-cluster with that of a **max-gap cluster** [1,7].

Definition 3. *Let $\theta \geq 1$. Let $V_C \subseteq V_S \cap V_T$ be a set of r vertices corresponding to genes all on the same chromosome χ_S in genome S and all on the same* *chromosome χ_T in genome T. Let $g_1, g_2, \ldots, g_r$ be a labelling of these genes according to their order on χ_S. Let $h_1, h_2, \ldots, h_r$ be a labelling of these same genes according to their order on χ_T. Let $p_S(\cdot)$ and $p_T(\cdot)$ indicate the positions of genes on χ_S and χ_T, respectively. Then if*

$$|p_S(g_{i+1}) - p_S(g_i)| \leq \theta \text{ and } |p_T(h_{i+1}) - p_T(h_i)| \leq \theta \tag{4}$$

for all $1 \leq i \leq r-1$, then V_C satisfies the max-gap criterion. If, in addition V_C is contained in no larger V_F also satisfying the criterion, then V_C is said to be a max-gap cluster.

Proposition 3. *Every ST-cluster with parameter θ satisfies the max-gap criterion with the same value of θ.*

Proof: Consider two successive genes in the ST-cluster in genome S. By Definition 2, they cannot be separated by more than $\theta - 1$ genes not in the cluster. Since this holds for all pairs of successive genes, both in S and in T, the max-gap criterion is met. □

The converse of Proposition 3 does not hold, however. The max-gap criterion only limits the number of **non-cluster elements** intervening, in either genome, between two cluster elements. Thus in the max-gap definition with $\theta = 2$, we could have a cluster $\{a, b, c, d, e, f\}$ with order $abcdef$ in S and $fbcdea$ in T, but this would not be an ST-cluster (though $\{b, c, d, e\}$ would be). Also, $a * bc$ in S and $bc * a$ in G_T could define a max-gap cluster $\{a, b, c\}$, where the asterisks represent genes not present, or remote, in S and T, respectively, but this would not be a ST-cluster (though $\{b, c\}$ would be).

3 Comparisons of Yeast Genomes

The data. The Yeast Gene Order Browser (YGOB) [3] contains complete gene orders and orthology identification among the five yeast species depicted in Figure 2: two descendents of an ancient genome duplication event, *Saccharomyces cerevisiae* and *Candida glabrata*, and three species that diverged before this event, *Ashbya gossypii, Kluyveromyces waltii* and *Kluyveromyces lactis.* For the ancient tetraploids, YGOB includes a reconstruction of the ancestral genome, which, with the help of further details supplied by Kevin Byrne and Jonathan Gordon (personal communication), allows us to identify duplicate genes as belonging to one of the two ancestral lineages, indicated by A and B in the figure, and to find two complete sets of clusters in each of these species, one in each lineage. For our purposes, then, the duplicate lineage effectively expands the data set from five to seven genomes.

Notation. With reference to Fig. 2 we will refer to the common ancestor of *Ashbya gossypii* and *Kluyveromyces lactis* as Node D, and to its immediate ancestor as Y. Nodes A and B will refer to the two ancestral lineages within both

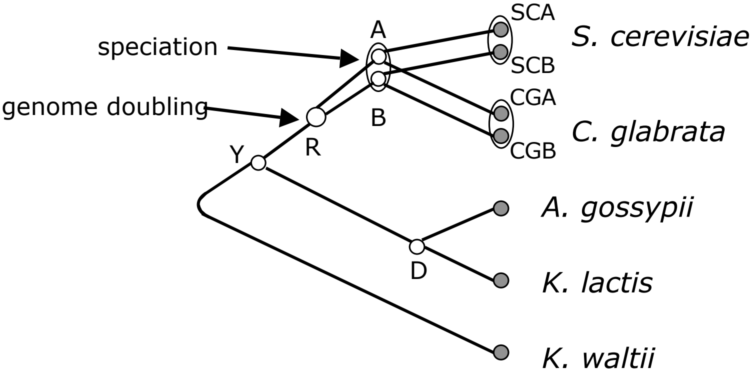


Fig. 2. Phylogeny of yeasts in YGOB. Whole genome doubling event at R giving rise to A and B lineages in *S. cerevisiae* (SCA, SCB) and *C. glabrata* (CGA, CGB) indicated, as is the speciation event at the divergence of these two species. Choice among the identified ancestor nodes Y, R, D, A or B to be the root is arbitrary in our mathematical analysis, but historically, the earliest divergence time is represented by the branching at the left of the phylogeny.

Saccharomyces cerevisiae and *Candida glabrata* at the time of speciation, while Node R will designate the tetraploid ancestral to these.

Defining lineage-specific clusters within a tetraploid descendant. The YGOB indicates the common ancestry, pre-speciation, in *Saccharomyces cerevisiae* and *Candida glabrata*, of two separate gene lineages, labelled A and B, in both genomes. To apply Definition 2, we first masked the identity of all lineage B genes without deleting them from their positions, and then applied the criterion to the lineage A genes to produce the edges in G_{SCA} and G_{CGA}. We then reversed roles of A and B, masking the identity of all lineage A genes without deleting them from their positions, and then applied the criterion to the lineage B genes to obtain G_{SCB} and G_{CGB}. Fig. 3 shows plots of the number of clusters detected as a function of θ, decreasing as a result of cluster amalgamation, featuring a distinct elbow near $\theta = 3$ for all the pairwise comparisons. This also shows a striking resemblance to the same analysis for max-gap clusters, suggesting that in these data, the max-gap clusters also satisfy our more stringent generalized adjacency criterion. In other contexts, perhaps in prokaryotes, more intense processes of local gene rearrangement may result in relatively more max-gap clusters.

In Fig. 3, we depict how cluster size is distributed and use this to assess the degree of relatedness of genomes or lineages.

4 Extensions to m Genomes

We can extend the definition of an ST-cluster based on two genomes S and T to an $ST \cdots U$-cluster based on the graphs $G_S, G_T, \ldots, G_U$ induced by the m

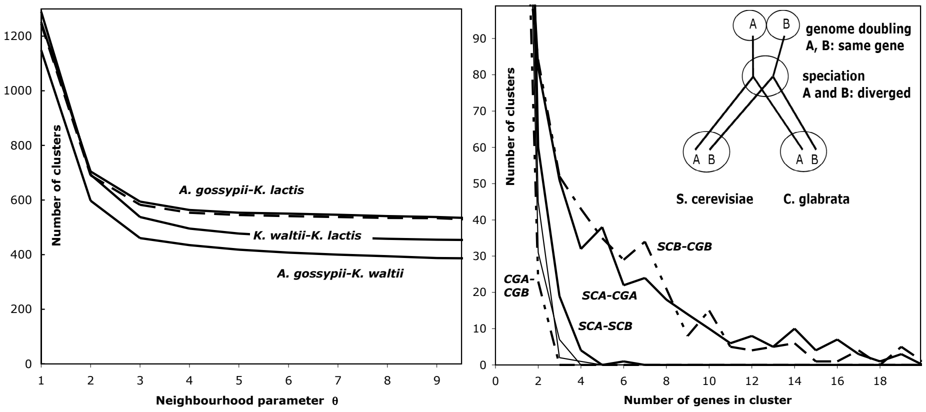


Fig. 3. Left: Dependence of number of clusters on neighbourhood parameter, showing that, independent of θ, *K. waltii* has fewer (larger) clusters when compared with the other two genomes, as might be expected from the closer phylogenetic relationship of the latter in Fig. 2. Dashed line indicates that the max-gap criterion returns fewer, larger clusters for the same value of θ — one max-gap cluster may contain several ST-clusters. Downward slope of all lines due to the incorporation of smaller clusters into larger ones as θ increases, demonstrating that almost all max-gap clusters also have conserved neighbourhood structure. Right: Distribution of size of clusters for $\theta = 2$, showing larger clusters, i.e., less evolutionary divergence, between same lineage SCA-CGA and SCB-CGB in different species than between different lineages. Also, the two different lineages are more diverged in CG than in SC, as confirmed for larger θ (not shown), consistent with the highly derived nature of the *C. glabrata* genome. Two thinner, unlabeled curves indicate SCA-CGB and SCB-CGA.

genomes $S, T, \ldots, U$. We simply extend Definition 1 to involve the intersection of the edge sets of m graphs instead of 2 graphs,

$$G_{ST \cdots U} = (V_S \cap V_T \cap \cdots \cap V_U, E_S \cap E_T \cap \cdots \cap E_U) \tag{5}$$

and then retain the set of vertices in each of the connected components of this graph as the $ST \cdots U$-clusters.

A more useful generalization turns out to involve the **median** of the m genomes $M_{ST \cdots U} = (V_S \cup V_T \cup \cdots \cup V_U, E)$, where E minimizes the sum of the sizes of the symmetric differences between E and the E_X. This is rapidly calculated using a majority rule. This graph may, however, sometimes not correspond to any genome, as we will discuss in Section 7. To verify that it does, we have to solve the fixed parameter version of the maximum bandwidth problem, which has a polynomial (but hard to implement) dynamic programming solution. Otherwise we can try to find the largest subset of E with bandwidth $\leq \theta$, which may require exponential search.

5 Optimizing Ancestral Nodes Minimizing Edge Appearances/Disappearances

Consider that the data at each terminal node consist of zeroes or ones, representing the presence or absence of each edge ϵ in that data genome. We wish to assign a zero or one for each edge at each ancestral node so as to minimize the number of times the presence/absence indicator changes value from one endpoint of an edge to the other, summed over all branches in the tree and summed over all edges ϵ. We will discuss this ancestral node optimization for unrooted binary trees, i.e., where each ancestral node has exactly three adjacent nodes, perhaps the simplest instance of dynamic programming on a tree [5, Chapter 2]. (This procedure is easily extended to non-binary trees.)

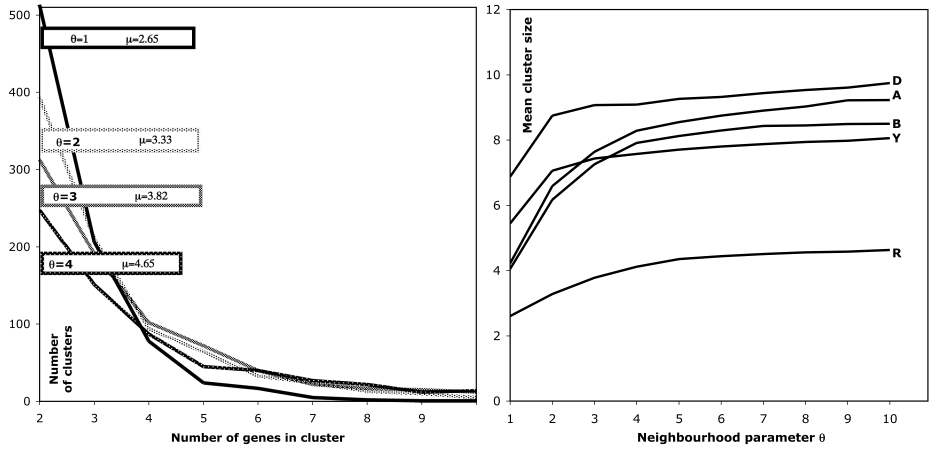


Fig. 4. Left: Distributions of cluster size, with mean μ, at Node R, for various values of θ. Smaller clusters amalgamate into larger ones as θ increases. Right: Mean cluster size at ancestral nodes, for various values of θ.

Dynamic programming requires two passes. In the forward pass, from the terminal nodes towards the root R (chosen arbitrarily from among the ancestor nodes, without consequences for the results), the value of the variable (the presence or absence of ϵ) may be established definitely at some ancestral nodes, while at other nodes it is left unresolved until the second, “traceback” pass, when any multiple solutions are also identified. We call those edges that are definitely present at a node the *optimals*, while those that are potentially present during the forward pass, the *near-optimals*. We need not discuss further those that are definitely excluded during the forward pass.

Note that the (arbitrary) designation of one ancestor node to be the root R determines, for each branch, which of its endpoints corresponds to the mother genome (the one proximal to the root), and which to the daughter (the one distal

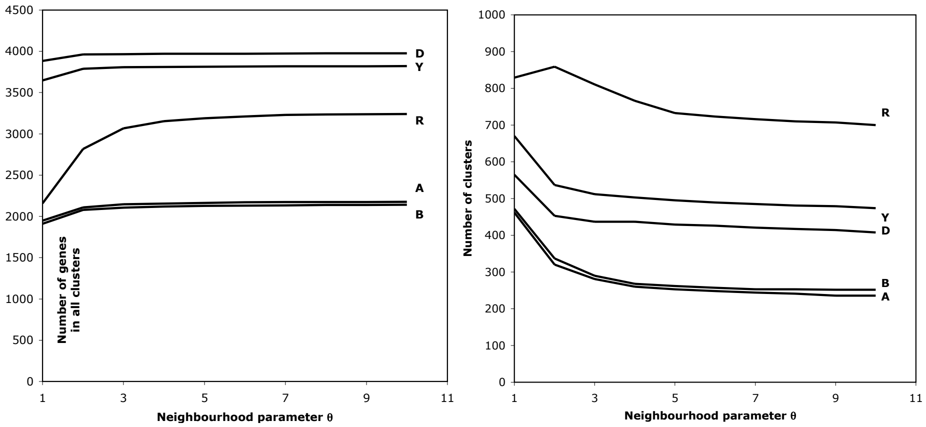


Fig. 5. Left: Total number of genes in clusters is remarkably stable, except for Node R, which recruits more genes up to $\theta = 4$. Right: As θ increases small clusters are amalgamated with larger ones, so that the total number of clusters decreases.

to the root). We order the nodes so that no node precedes any of its daughters. (This is always possible for a rooted tree.)

Suppose ancestral node N (other than the root R) has daughter nodes K and H. Because of the way we have ordered the nodes, by the time we reach N during the forward pass, we have already decided, for each daughter, whether ϵ is an optimal or near-optimal. Then if ϵ is optimal for both K and H, then it is optimal for N. If it is optimal for only one of K and H, it is near-optimal for N. For the root node R, with three daughters, if ϵ is optimal for at least two of the three, then it is optimal for R. We need not consider near-optimals for R.

For the traceback, reversing direction in the same order, starting at R, if ϵ is optimal for a mother node and near-optimal for its daughter, then ϵ is promoted to optimal status in the daughter. (This operationalizes the "majority rule" mentioned in Section 4.)

Note that in this method, the presence or absence of genes in the ancestral genomes derives solely from the presence or absence of at least one edge having that gene as an endpoint.

6 Gene Clusters at the Ancestral Nodes of the Yeast Phylogeny

6.1 Cluster Statistics

Introducing the generalized adjacencies through the neighbourhood parameter allows clusters to be conserved despite local rearrangements. This is seen in Fig. 4, where the distribution of cluster sizes (number of vertices) at Node R is seen to spread out to larger values as θ increases.

The average sizes of clusters is much higher in the other ancestral nodes, though they follow the same trend, as is seen on the right of Fig. 4.

While the average cluster size increases, the number of genes involved in these clusters at a given node does not change much, as seen in Fig. 5. Consequently, as seen on the right of the figure, the number of clusters drops.

6.2 Evolution and Cluster Coherency

From node to node the number of clusters and the genes they contain change. We can, however, assess to what extent this change is gradual or abrupt. If a cluster simply gains or loses a few genes, or if a cluster divides in two, or if two merge to become one, we may consider the resulting configuration a gradual change. We operationalize this by saying two clusters, one in each of two ancestral genomes, are in conflict unless one is nested in the other or they are disjoint. In Table 1, we show what proportion of each ancestor's clusters are in conflict with their adjacent nodes' clusters.

Table 1. Conflicts in clusters beteeen genomes at two ends of each tree branch, as a function of θ. Percentage conflict out of the total number of clusters in genome in left hand column.

		Neighbourhood parameter		
Node	Adjacent Node	1	3	8
A	R	20	36	37
B	R	23	36	40
R	A	10	16	16
R	B	11	16	17
R	Y	0	0	0
D	Y	0	1	1
Y	D	0	0	1
Y	R	0	1	1

Thus cluster evolution has been exceedingly gradual among the diploid genomes, but a good proportion of the A and B lineage clusters are seriously disrupted in their common ancestor.

7 Bandwidth of the Clusters

We have constructed clusters of genes based on adjacencies presumed to have been present in the ancestral genomes. While these are most parsimonious inferences, they are not sufficient to reconstruct the entire genomes, mainly because we have tried to compute neither how to partition the clusters among chromosomes nor how to impose a linear order within a cluster. Indeed, the dynamic programming is not even able to ensure that the clusters are compatible with the generalized adjacency structure imposed on the data genomes in Definition 2, for the reasons alluded to in Section 4. In other words, there is no constraint on the connected components, and hence the entire graph inferred at an ancestral node, to have maximum bandwidth $\leq \theta$. If the bandwidth is larger, it means that we can construct no genome where the vertices in the connected component in question can be linearly disposed so that each edge has less than θ genes intervening between the two endpoints.

On the other hand, there is no compelling reason to insist on this bandwidth restriction on the ancestral genomes. Our initial goal was to find how clusters of vertices are preserved or evolve along various evolutionary lineages, and if the bandwidth is larger at some ancestor, this simply suggests that the cluster was looser at that time.

Whatever the importance or the interpretation we attach to bandwidth, it is thus of great importance to see how it is preserved or changed in the ancestral genomes we are investigating.

The problem of inferring the maximum bandwidth of a graph is, however, not trivial. Indeed, it is NP-complete [9], though Saxe [10] showed that detecting whether bandwidth is greater than θ is of polynomial complexity. Unfortunately, we have no implementation of Saxe's dauntingly high-level pseudo-code; in any case we are more interested in knowing the bandwidth than in testing whether it is greater than θ.

Thus we are led to investigate the many heuristics available for estimating the bandwidth. For example if there is a vertex of degree $> 2\psi$, the bandwidth must be greater than ψ, as is clear from the upper bound discussed in Theorem 1. Many heuristics emanate from an interest in reducing bandwidth in matrix theory. For sparse matrices, the best-known method is the Cuthill-McKee method [4] and its modification, the reverse Cuthill-McKee (RCM) algorithm [6,8]. We have implemented the latter to study the bandwidth of the components we have reconstructed at the ancestral nodes. Since the results of RCM depend on the input order of the vertices, we ran the algorithm 100 times with different orders to find the minimum estimate for the bandwidth, as displayed in Table 2. In all but three of the thirty entries in the table, the value shown was already detected after 10 runs.

As can be seen, the bandwidth exceeds θ in most cases. Inspection of the graphs show that this is due to a small number of vertices of high degree. If we wanted to constrain the ancestral genome graphs to have maximum bandwidth $\leq \theta$, we could:

- After the dynamic programming, test each node for bandwidth and exclude one or more vertices from the graph. It would not seem appropriate, however, to exclude the vertices of highest degree, since these are likely to be the most central to the cluster. Rather we would exclude some vertices of low degree adjacent to vertices of highest degree.
- As a less *ad hoc* solution, during the traceback of the dynamic programming, ensure that the set of edges being constructed never exceeds bandwidth θ. This may be a complex undertaking, however, since it may require testing all subsets of the set of near optimal edges eligible to promotion to optimal

Table 2. Minimum out of 100 runs of the RCM algorithm applied to edge sets produced by dynamic programming at the ancestral nodes of yeast evolutionary tree for various values of the neighbourhood parameter θ.

θ	Node				
	A	R	B	Y	D
1	1	1	1	1	1
2	3	4	2	3	2
3	4	5	4	5	5
4	5	7	5	7	5
5	6	7	6	8	7
6	9	8	7	8	8
7	9	10	9	9	9
8	9	11	10	11	11
9	13	13	11	10	11
10	13	13	11	16	17

status. Note that this approach potentially interferes with the exactness of the dynamic programming.

- Intervene directly in the dynamic programming recurrence. To conserve exactness, this approach would require storing and searching over a structure more complex than just the sets of optimals and near optimals.

8 Conclusions

The generalized adjacencies we have introduced allow us to recognize clusters even though they have been perturbed by local rearrangements. That the maxgap criterion gives approximately the same number of clusters means that maxgap is too weak a criterion in this context in that it doesn't recognize order conservation in the clusters.

Our separation of the A and B lineages as separate phylogenetic lineages is validated by the higher number of within-lineage clusters than within-species clusters, with the *C. glabrata* genome appearing highly rearranged.

We have shown the interplay of bandwidth considerations and the dynamic programming optimization of ancestral nodes in a given phylogeny. There is scope for improving our estimate of bandwidth, perhaps with approximation algorithms such as the semi-definite-programming approach in [2].

The neighbourhood parameter allows us to control the distribution of cluster sizes and the number of clusters. It allows us to explore the trade-off between the size of clusters and the rate of conflict between clusters in connected ancestral nodes.

Acknowledgments

We thank Ken Wolfe, Kevin Byrne and Jonathan Gordon for encouragement and for valuable information, including key data beyond the inestimable resource they have made available to the yeast and genomics communities. We also thank Howard Bussey for a helpful discussion and Robert Warren for his sustained collaboration in the polyploid genomics project. Research supported in part by a grant to DS from the Natural Sciences and Engineering Research Council of Canada (NSERC). DS holds the Canada Research Chair in Mathematical Genomics.

References

1. Bergeron, A., Corteel, S., Raffinot, M.: The algorithmic of gene teams. In: Guigó, R., Gusfield, D. (eds.) WABI 2002. LNCS, vol. 2452, pp. 464–476. Springer, Heidelberg (2002)
2. Blum, A., Konjevod, G., Ravi, R., Vempala, S.: Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems. In: Proceedings of the 30th ACM Symposium on the Theory of Computing, pp. 95–100 (1997)
3. Byrne, K.P., Wolfe, K.H.: The Yeast Gene Order Browser: combining curated homology and syntenic context reveals gene fate in polyploid species. Genome Research 15, 1456–1461 (2005)
4. Cuthill, E., McKee, J.: Reducing the bandwidth of sparse symmetric matrices. In: Proceedings of the 24th National Conference of the ACM, pp. 157–172 (1969)
5. Felsenstein, J.: Inferring phylogenies. Sinauer Associates, Sunderland, MA (2004)
6. George, A.: Computer implementation of the finite element method, STAN-CS-71-208, Computer Science Dept., Stanford Univ., Stanford, CA (1971)
7. Hoberman, R., Sankoff, D., Durand, D.: The statistical analysis of spatially clustered genes under the maximum gap criterion. Journal of Computational Biology 12, 1081–1100 (2005)
8. Liu, J., Sherman, A.: Comparative analysis of the Cuthill-Mckee and the reverse Cuthill-Mckee ordering algorithms for sparse matrices. SIAM Journal of Numerical Analysis 13, 198–213 (1975)
9. Papadimitriou, C.H.: The NP-completeness of the bandwidth minimization problem. Computing 16, 263–270 (1976)
10. Saxe, J.: Dynamic-programming algorithms for recognizing small-band-width graphs in polynomial time. SIAM Journal of Algebraic and Discrete Methods 1, 363–369 (1980)