



Fast identification and statistical evaluation of segmental homologies in comparative maps

Peter P. Calabrese¹, Sugata Chakravarty² and Todd J. Vision^{3,*}

¹Department of Mathematics, University of Southern California, Los Angeles, CA 90089, USA, ²Department of Operations Research, and ³Department of Biology, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA

Received on January 6, 2003; accepted on February 20, 2003

ABSTRACT

Motivation: Chromosomal segments that share common ancestry, either through genomic duplication or species divergence, are said to be *segmental homologs* of one another. Their identification allows researchers to leverage knowledge of model organisms for use in other systems and is of value for studies of genome evolution. However, identification and statistical evaluation of segmental homologies can be a challenge when the segments are highly diverged.

Results: We describe a flexible dynamic programming algorithm for the identification of segments having multiple homologous features. We model the probability of observing putative segmental homologies by chance and incorporate our findings into the parameterization of the algorithm and the statistical evaluation of its output. Combined, these findings allow segmental homologies to be identified in comparisons within and between genomic maps in a rigorous, rapid, and automated fashion.

Availability: <http://www.bio.unc.edu/faculty/vision/lab/>

Contact: tjv@bio.unc.edu

Keywords: homology, comparative maps, syntenic, genome evolution

INTRODUCTION

Multiple biological features that are descended from a single common ancestor are said to be *homologous* to one another. *Comparative mapping* involves the identification of homologous features among genomic maps. Between distantly related organisms, the most commonly used features for comparative maps are protein coding genes, both because of their ubiquity and because of the ability of local alignment search tools to detect the relationship among highly diverged protein sequences.

When multiple pairs of homologous features appear in roughly colinear order in two genomic segments, it suggests that the order itself was inherited from a common

ancestor. Two such segments are called *segmental homologs* (SH). When dealing with an incompletely mapped genome, knowing that two segments are homologous is useful in that it suggests that other (unmapped) features within those same segments may have homologous counterparts in the opposite segment. However, the distinction between homologous features and homologous segments is not an absolute one. For our purposes, it is convenient to understand a feature as being an interval defined by a single protein-coding gene or a small family of physically clustered and closely related protein coding genes, while a segment consists of multiple such features.

Many SH have been reported for related sets of species using dense genomic maps composed of homologous markers. Well-known examples of comparative maps include that of human and mouse (<http://www.ncbi.nlm.nih.gov/Homology/>) and the major species of cereal grains (<http://www.gramene.org>). Historically, SH in these maps have been identified using *ad hoc* methods. Though feasible for one-time analysis of highly similar genomes, such methods tend to be slow, not fully reproducible, not subject to statistical scrutiny, and not sufficiently sensitive to detect highly diverged SH. The particular difficulties of highly divergent SH include the following.

1. Nucleotide substitutions obscure homology between many pairs of features at the sequence level.
2. Rearrangements such as inversions and translocations subdivide one SH into multiple smaller SH, each containing fewer homologous features.
3. Feature content diverges among homologous segments over time due to gene loss and transposition. Gene loss is especially frequent after genomic duplication events (Ku *et al.*, 2000; Wolfe, 2001). Thus, segments may contain many features that do not have counterparts in their homologs.
4. Minor rearrangements shuffle the relative ordering and orientation of features within each homolog (Seioche *et al.*, 2000). This makes it necessary to

*To whom correspondence should be addressed.

look for a less-than-perfect linear correspondence in the order and orientation of homologous features.

5. Individual genes appear to be duplicated at a very high frequency, particularly in eukaryotes (Lynch and Conery, 2000). Thus, single features may have many homologs, only a fraction of which are due to segmental homology.

A method for the identification of divergent SH must take these considerations into account. The ability to identify such SH would be particularly useful for comparative map analyses of the growing number of complex, eukaryotic genomes of economic and scientific importance for which there now exist dense *transcript maps*, each detailing the relative positions of hundreds to thousands of protein-coding genes.

An important problem which needs to be addressed is the statistical evaluation of putative SH. How often would suggestive patterns arise by chance in the absence of SH but in the presence of large numbers of non-segmental feature homologies? Recently, permutation tests have been used to control for false-positives in the identification of SH (Vision *et al.*, 2000; Gaut, 2001). However, this approach is computationally expensive and does not permit very precise estimation of the probabilities of rare events. Thus, a more formal statistical framework for the identification of SH would be desirable (Durand and Sankoff, 2002).

SYSTEM AND METHODS

Each genome to be compared can be thought of consisting of one or more linear sequences of features, called *contigs*. We assume a comparison between single contigs (e.g. unichromosomal genomes) in what follows, but extension to multiple contigs is straightforward. A feature is typically a protein-coding gene but may be any entity to which it possible to ascribe homology to other features. We assume that the distance between adjacent features on a contig is always one unit. One can visualize the comparative mapping data in the form of a matrix. If two features are homologous, then there is a *point*, represented by a one, at the intersections of the row and column indexed by those features. If not, there is a zero.

When two segments are homologous, we expect them to share multiple homologous features in approximately colinear order. In the matrix, this would appear as a clump of closely spaced points in a roughly diagonal line (Fig. 1). When we encounter such a clump, we take it as evidence for SH between the intervals defined by the two pairs of points that are most distant within each contig. The problem we face is that, in real data, many, if not most, of the points in the matrix may not be part of larger *segmental* homologies. We need to be able to discern when a clump

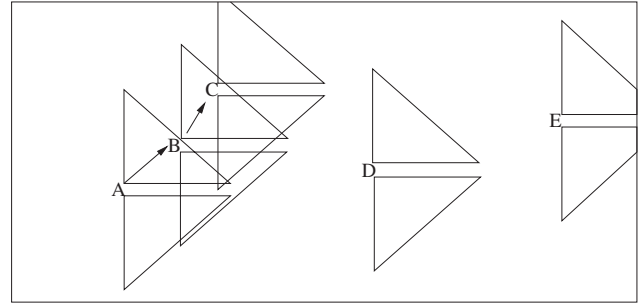


Fig. 1. A 3-clump (consisting of points A, B and C) suggestive of segmental homology. The neighborhood of A contains point B and the neighborhood of B contains point C, but the neighborhoods of C, D and E contain no points. Here, neighborhoods are defined by Manhattan distance. The neighborhood of C is restricted by the top boundary and that of E by the right boundary of the matrix.

of closely spaced points is unlikely to have occurred by chance.

We propose a simple null model for homologies among individual features in the absence of segmental homology and a definition of what constitutes an exceptional clump suggestive of SH. Together, these allow us to calculate the expected number of clumps of a given size that we expect simply by chance.

Consider an $r \times c$ matrix, where each entry is independently a one with probability h and a zero with probability $(1 - h)$. If an entry is a one, we will call that entry a point. We are thinking of a large, sparse matrix. For each entry x , we define a neighborhood T_x . The only restriction on T_x is that all entries are to the right of x . We define a k -clump as a set of points $\{x_1, x_2, \dots, x_k\}$ such that

1. $x_i = 1$ for $i = 1, 2, \dots, k$
2. $x_{i+1} \in T_{x_i}$ for $i = 1, 2, \dots, k - 1$
3. There are no points to the left of x_1 which contain x_1 in their neighborhood.

We call x_1 the left-endpoint of the k -clump. Let n be the number of entries in T_x , and p be the probability T_x contains at least one point.

$$p = 1 - (1 - h)^n \quad (1)$$

Define the diameter d as the maximum over $y \in T_x$ of the larger of the number of rows or columns between y and x . We call a clump of size k or greater a kg -clump.

We want to calculate the probability distribution for the number of kg -clumps. If several such clumps intersect, we only count their intersection once. We calculate an upper and lower bound and use the Chen-Stein Poisson

approximation (Arratia *et al.*, 1990), which provides explicit error bounds. This model is an example of a coverage process (e.g. Hall *et al.*, 1988).

First we consider the probability an entry x , which is sufficiently far from the boundary, is the left endpoint of a kg -clump. The probability that x is a point and that there are no points to the left containing x in their neighborhood is $(1 - p)h$.

First, we consider the upper bound. There are n^{k-1} sets of entries that, were they all points, would be a kg -clump with x as their left endpoint. For each set, the probability that it contains all points is h^{k-1} . So, an upper bound for the probability that one of these sets contains all points is $(nh)^{k-1}$. An upper bound for x being the left endpoint of a kg -clump is

$$p'_u = (1 - p)h(nh)^{k-1} \quad (2)$$

Next, we consider a lower bound. The calculation in the previous paragraph was for the expected number of kg -clumps with x as their left endpoint and it ignored the possibility that more than one clump may intersect. We consider k -clumps $\{x_1, x_2, \dots, x_k\}$ with the additional properties, for $i = 2, \dots, k$

1. x_i is the only non-zero entry in $T_{x_{i-1}}$.
2. x_{i-1} is the only point to the left of x_i which contains x_i in its neighborhood.

The number of such restricted-definition k -clumps is less than the number of regular k -clumps and therefore provides a lower bound for the probability that x is the left endpoint of a regular kg -clump

$$p_l = (1 - p)h[nh(1 - p)^2]^{k-1} \quad (3)$$

For entries near the boundary, the calculations above are not correct because there are fewer neighboring entries to consider. For x in the left d columns, bottom d rows, or top d rows, the probability that there are no points that contain x in their neighborhood is greater than $(1 - p)$. For an upper bound, substitute one for this probability, and revisiting Equation (2) define a new upper bound

$$p_u = h(nh)^{k-1} \quad (4)$$

For x contained in the right $(k - 1)d$ columns, bottom $(k - 1)d$ rows, or top $(k - 1)d$ rows, the probability that x is the left endpoint of a kg -clump is less than calculated above. For a lower bound, substitute zero for this probability.

Above, we have calculated bounds for the probability an entry is the left endpoint of a kg -clump. What we want to calculate is the number of such clumps in the

matrix. If two entries are sufficiently far apart, whether one entry is a left endpoint of a kg -clump is independent of whether the other entry is a left endpoint. However, since we only count intersecting clumps once, for close entries this independence is not true. We apply Theorem 1 of (Arratia *et al.*, 1990). For x , the dependence set is the square of width $2kd$ centered at x . Define

$$b_u = (rc)(2kdp_u)^2 \quad (5)$$

$$b_l = [r - 2(k - 1)d][c - (k - 1)d](2kdp_l)^2 \quad (6)$$

For the upper bound, the number of kg -clumps is approximately Poisson with mean $(rc)p_u$ and total variation error less than $4b_u$. So, a conservative p -value for there to be any k -clumps in the matrix is

$$1 - \exp(-rcp_u) \quad (7)$$

For the lower bound, the number is approximately Poisson with mean

$$[r - 2(k - 1)d][c - (k - 1)d]p_l \quad (8)$$

and total variation error less than $4b_l$.

Above we have considered a matrix for the case where we are comparing two different genomes. When we are comparing one genome to itself, the matrix is symmetric, and we are only interested in the half above the diagonal. The analysis is similar, and the conservative p -value for there to be any kg -clumps is

$$1 - \exp\left(-\frac{rcp_u}{2}\right) \quad (9)$$

ALGORITHM

Under the null model, we can calculate the probability of observing a clump as a function of the number of points it contains. Here, we express this as a simple scoring function that can be used in a dynamic programming algorithm for finding all maximal k -clumps in the matrix. Imagine a directed acyclic graph G in which the the set of vertices V are the points in the matrix and edges $E(i, j)$ extend from each point $i \in V$ to all points $j \in T_i$. The score on each edge s_{ij} is one, and the score on a clump S is the sum of the scores on each edge. Thus, the score of a clump is simply the number of points in the clump. We can find all maximal k -clumps in the matrix by recursion since the score of the clump terminating at j is

$$S_j = \max(S_i + s_{ij}) \text{ for all } i \text{ such that } j \in T_i \quad (10)$$

In practice, one might wish to set a minimum score based on conservative p -values from Equation (7) and only report clumps with $S_j > S_{min}$. The following algorithm creates a traceback graph H in which the clumps are the connected components.

Algorithm: Find k -clumps

- Step 1** Sort the points in topological order (Sedgewick, 1990).
- Step 2** For each point j , calculate S_j using Equation (10). If no points include j in their neighborhood, then $S_j = 1$. If $S_j > 1$, construct an edge in H from j to its predecessor of maximal score.
- Step 3** For each vertex in H with outdegree zero, collect all vertices in that connected component and report it as a k -clump.

IMPLEMENTATION

FISH (Fast Identification of Segmental Homology) is a software package, written in C++, that implements the maximal k -clump finding algorithm described above and reports the pertinent statistics for each clump and pair of adjacent points. It requires as input a list of the linear order and orientation of features on each contig and a list of the pairwise homologies between features. It employs the theory from **System and Methods** both to parameterize the dynamic programming algorithm and to statistically evaluate its output. Here we describe a number of implementation details that we believe to be important in the analysis of real data.

Neighborhood size and shape

Neighborhood size determines how likely a point is, under the null model, to have a predecessor for a given h . One can use Equation (1) to choose a neighborhood with a suitable value of p , the probability that T_x contains a point, depending on whether one wishes to detect clumps with few closely spaced points, or clumps with a larger number of distantly spaced points. The former would be appropriate when analyzing two genomes that have undergone many large-scale chromosomal rearrangements, while the latter would be more appropriate when searching for anciently duplicated chromosomal segments within a single genome.

The null model that we describe puts few restrictions on the geometry of the neighborhood. This is convenient in that it allows us to define neighborhood geometries that permit k -clumps in which the points are not perfectly colinear in the two segments. In general, point j is included in the neighborhood of point i if,

1. j is in a column to the right of i
2. j is not in the same row as i
3. For point i , having coordinates (i_x, i_y) , and point j , having coordinates (j_x, j_y) , the distance between i and j is less than some critical value d_c .

In FISH, d_c is the largest Manhattan distance $d_M = |i_x - j_x| + |i_y - j_y|$ for which the value of p is below some user-defined threshold. But any distance measure may be employed in its stead. More study of the spacing between adjacent points in actual data would be helpful in determining the most appropriate measure of distance for defining neighborhoods.

Multiple predecessors

On occasion, a single point may have multiple predecessors that confer the same edge score. In order to avoid left-branching clumps, which make little biological sense, it is necessary to choose only one predecessor. This is achieved in FISH by giving every entry within a neighborhood, and thus each potential edge, a unique rank. The predecessor having the edge of lowest rank is the one that is chosen. In FISH, the default *ad hoc* ranking procedure balances several considerations: predecessors should be close by one of the distance measures above, the distance along the two axes should be close to symmetric (in the case of Manhattan distance) and the edge should minimize departure from colinearity within a clump. The rare ties that remain are broken randomly. The ranking procedure can be varied to suit different biological assumptions or applications. Note that this does not ensure the absence of right-branching clumps, which FISH deals with in an *ad hoc* post-processing step.

Feature orientation

In some, though not all, comparative mapping datasets, it is also possible to consider the orientation of the homologous features in the two different segments (e.g. the transcriptional direction of protein coding genes). Two homologies within the same clump are expected to maintain one of the three canonical orientations relative to one another in the absence of inversions (Fig. 2). Let i be a point representing homology between features i_x and i_y , while j is a point representing homology between features j_x and j_y . Let each feature have an orientation θ of -1 or 1 . If two points i and j are in canonical orientation, then $\theta_{i_x}\theta_{i_y} = \theta_{j_x}\theta_{j_y}$. The probability of two adjacent points in a k -clump being in canonical orientation under the null model is rather small, only 0.25. Small inversions do appear to be frequent in eukaryotic (if not prokaryotic) genomes (Seioghe *et al.*, 2000; Huynen *et al.*, 2001), so some adjacent points in non-canonical orientation are to be expected. In a pair of segments that are homologous, however, adjacent points showing canonical orientation will tend to occur more often than expected by chance.

Data preprocessing

For real datasets from complex genomes, a number of pre-processing steps are desirable (Vision and Brown, 2000). The first step, which we call *detandemization*, consists of

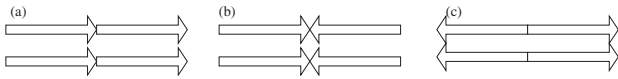


Fig. 2. The three canonical orientations for two pairs of homologous features: parallel (a), convergent (b) and divergent (c). Homologous features are aligned vertically, neighboring genes on the same contig adjoin end to end.

collapsing tandem and near-tandem arrays of homologous features into single composite features. The reason for this is that there are typically many such clusters of closely related genes on eukaryotic chromosomes (e.g. Kihara and Kanehisa, 2000). Such tandem arrays pose a problem since they can create clumps of points in the matrix in the absence of segmental homology. Detandemization prevents such clumps from appearing in the output. It also serves to reduce the number of points in the matrix and correspondingly increase the search neighborhood size (or decrease the minimum k -clump length) for the same level of Type I error. As a result, though detandemization will tend to reduce the power of the algorithm for detecting SH that are predominantly composed of homologous tandem arrays, it can substantially increase the power for detecting those that are not.

The second preprocessing step is to filter the list of feature homologies. In complex genomes, there is much variation in size among gene families, with some families having only one member and others hundreds (Sonnhammer and Durbin, 1997). The few large families contribute an inordinate proportion of the homologies in the unprocessed matrix since the number of matches is proportional to the square of the family size. The inclusion of all the homologies from these large families would unnecessarily restrict the neighborhood size by inflating the value of h . To avoid this, FISH can filter the dataset by ranking the homologs of each feature by some user-input criterion (such as the extent of divergence between homologous protein sequences) and discarding those of low rank. It can be implemented in such a way as to enforce the symmetry of the homology matrix. Implementation details for both preprocessing steps are described in the FISH documentation.

RESULTS

Simulations under the null model

We have simulated the null model and compared the observed results with the theoretical bounds. The parameters correspond to those used in the comparison of *Arabidopsis thaliana* chromosomes 2 and 4 discussed below: $r = 3730$, $c = 3825$, there are 948 points, and so $h = 948/(rc) = 6.64 \times 10^{-5}$. The Manhattan metric with

Table 1. In simulated data, the sample mean and standard error of the number of kg -clumps, and the theoretical upper and lower bounds for the mean

k	sample mean	standard error	upper bound	lower bound
2	45.8	0.06	47.6	40.1
3	2.28	0.02	2.39	1.78
4	0.113	0.003	0.120	0.079
5	0.006	0.001	0.006	0.004
6	0.0003	0.0002	0.0003	0.0002

$d_c = 29$ defines the neighborhood. Under the null model, the probability there is a point in this neighborhood is 0.049. In 10 000 simulated matrices, no clumps larger than seven were observed. Table 1 shows that the theoretical calculations provide excellent bounds on the simulated observations.

Analysis of chromosomal duplications in *Arabidopsis thaliana*

Table 2 shows the results for an actual comparison of *A.thaliana* chromosomes 2 and 4, along with the calculated p -values. The matrix can be viewed online at http://www.bio.unc.edu/faculty/vision/lab/arab/science_supplement/chr2v4.gif. Analysis was done using two neighborhoods: one as above for the simulations with $p = 0.049$, and another with $d_c = 14$, which gives $p = 0.013$. In both cases, there are several clumps that are highly significant under the null model. Note that the conservative p -value and the lower bound are quite close, and that the total variation error is small. For more detailed analyses of this dataset, see Simillion *et al.* (2002); Vision *et al.* (2000).

DISCUSSION

Further biological considerations

An assumption underlying our model, one that is often implicit in the literature (Gaut, 2001; Vandepoele *et al.*, 2002), is that the probability of homology between any two features is independent of the positions of those two features provided the segments themselves are not homologous. Is this a valid assumption? Feature homology in the absence of segmental homology implies that one of the homologous features has been duplicated and/or transposed in one or both of the genomes (or that rearrangements have shuffled gene order to the point of randomness). It follows that the null model is only correct when single features are duplicated and/or transposed to random positions in the genome. The duplication scenario is not violated by tandem duplications provided that detandemization is performed (see **Implementation**). However, there is empirical evidence that the process of

Table 2. In *Arabidopsis* chromosomes 2 v. 4, for two neighborhood sizes, the observed k -clumps and the conservative p -value, its lower bound, and the total variation error in this calculation

k	# obs.	cons. p -value	lower bound	total var. error
$p = 0.049$				
7	1	1.52×10^{-5}	6.93×10^{-6}	9.29×10^{-12}
9	1	3.84×10^{-8}	1.37×10^{-8}	9.78×10^{-17}
10	1	1.93×10^{-9}	6.05×10^{-10}	3.05×10^{-19}
14	1	1.23×10^{-14}	2.33×10^{-15}	2.42×10^{-29}
16	1	3.10×10^{-17}	4.58×10^{-18}	2.01×10^{-34}
19	1	3.93×10^{-21}	3.96×10^{-22}	4.56×10^{-42}
20	1	1.96×10^{-22}	1.75×10^{-23}	1.28×10^{-44}
22	1	4.98×10^{-25}	3.41×10^{-26}	9.83×10^{-50}
26	1	3.17×10^{-30}	1.29×10^{-31}	5.57×10^{-60}
58	1	8.57×10^{-72}	2.78×10^{-75}	2.02×10^{-142}
$p = 0.013$				
5	1	1.09×10^{-5}	9.59×10^{-6}	4.84×10^{-13}
6	2	1.13×10^{-7}	9.64×10^{-8}	7.48×10^{-17}
7	2	1.18×10^{-9}	9.69×10^{-10}	1.09×10^{-20}
8	3	1.22×10^{-11}	9.74×10^{-12}	1.53×10^{-24}
9	2	1.26×10^{-13}	9.80×10^{-14}	2.09×10^{-28}
10	2	1.31×10^{-15}	9.84×10^{-16}	2.77×10^{-32}
11	1	1.36×10^{-17}	9.89×10^{-18}	3.60×10^{-36}
14	1	1.51×10^{-23}	1.00×10^{-23}	7.23×10^{-48}
18	1	1.75×10^{-31}	1.02×10^{-31}	1.59×10^{-63}

transpositional gene duplication has a slight tendency to leave the copy at a position closer to the site of origin than would be expected by chance (e.g. Vision *et al.*, 2000). As a result, our method may underestimate the null frequency of kg -clumps that involve nearby segments in a genome self-comparison and overestimate the null frequency of clumps involving distant segments. This bias appears to be small, but it is shared by all the current statistically-based methods for identification of SH, including those based upon permutation tests, and it warrants further study.

Comparison to other methods

A number of other computational approaches for identifying and evaluating SH have been proposed recently (Delcher *et al.*, 1999; Durand and Sankoff, 2002; Fujibuchi *et al.*, 2000; Gaut, 2001; Goldberg *et al.*, 2000; Vandepoele *et al.*, 2002). The method described here has a number of attributes which make it particularly appropriate for the identification of highly diverged SH in large and complex genomes.

1. It is sensitive to the presence of clumps even when they account for only a small fraction of the feature homologies in the matrix. Popular methods for fast alignment of whole genomes (e.g. Delcher *et al.*, 1999) rely on the presence of unique sequence

matches, which may not be suitable for use with complex genomes having a high frequency of single-gene duplication.

2. It does not strictly enforce colinearity among the homologous features in the two segments. This is important, since small inversions appear to be commonplace in eukaryotic genomes (Seioche *et al.*, 2000).
3. The dynamic programming algorithm coupled with the analytic formulae in **System and Methods** allow putative SH to be identified and statistical results to be evaluated extremely quickly. The running time and memory usage of the algorithm both scale approximately linearly with the number of points in the matrix. Comparison of all five *A.thaliana* chromosomes with each other using FISH takes approximately five seconds on a P3 processor, most of which is devoted to file handling.
4. The hands-off nature of the algorithm allows it to be incorporated into an automated analysis pipeline provided appropriate parameters have been previously selected.

ACKNOWLEDGEMENTS

We wish to thank B. Gaut, N. Rosenberg and M. Waterman for helpful discussions. This work was supported by NSF DMS-0102008 to PPC and NSF DBI-40734 to TJV.

REFERENCES

- Arratia,R., Goldstein,L. and Gordon,L. (1990) Poisson approximation and the Chen-Stein method. *Statistical Science*, **5**, 403–424.
- Delcher,A.L., Kasif,S., Fleischmann,R.D., Peterson,J., White,O. and Salzberg,S.L. (1999) Alignment of whole genomes. *Nucleic Acids Res.*, **27**, 2369–2376.
- Durand,D. and Sankoff,D. (2002) Tests for gene clustering. *RE-COMB 2002*, 144–154.
- Fujibuchi,W., Ogata,H., Matsuda,H. and Kanehisa,M. (2000) Automatic detection of conserved gene clusters in multiple genomes by graph comparison and P-quasi grouping. *Nucleic Acids Res.*, **28**, 4029–4036.
- Gaut,B.S. (2001) Patterns of chromosomal duplication in maize and their implications for comparative maps of the grasses. *Genome Res.*, **11**, 55–66.
- Goldberg,D.S., McCouch,S. and Kleinberg,J. (2000) Algorithms for constructing comparative maps. In Sankoff,D. and Nadeau,J.H. (eds), *Comparative Genomics*. Kluwer, Dordrecht, pp. 243–261.
- Hall,P. (1988) *Introduction to the Theory of Coverage Processes*. Wiley, New York.
- Huynen,M.A., Snel,B. and Bork,P. (2001) Inversions and the dynamics of eukaryotic gene order. *Trends Genet.*, **17**, 304–306.
- Kihara,D. and Kanehisa,M. (2000) Tandem clusters of membrane proteins in complete genome sequences. *Genome Res.*, **10**, 731–743.

- Ku, H.-M., Vision, T.J., Liu, J. and Tanksley, S.D. (2000) Comparing sequenced segments of the tomato and *Arabidopsis* genomes: large-scale duplication followed by selective gene loss creates a network of synteny. *Proc. Natl Acad. Sci. USA*, **97**, 9121–9126.
- Lynch, M. and Conery, J. (2000) The evolutionary fate and consequences of duplicate genes. *Science*, **290**, 1151–1155.
- Sedgewick, R. (1990) *Algorithms in C*. Addison-Wesley, Reading, MA, pp. 479–481.
- Seoighe, C., Federspiel, N., Jones, T., Hansen, N., Bivolarovic, V., Surzycki, R., Tamse, R., Komp, C., Huizar, L., Davis, R.W. et al. (2000) Prevalence of small inversions in yeast gene order evolution. *Proc. Natl Acad. Sci. USA*, **97**, 14433–14437.
- Simillion, C., Vandepoele, K., Van Montagu, M.C., Zabeau, M. and Van De Peer, Y. (2002) The hidden duplication past of *Arabidopsis thaliana*. *Proc. Natl Acad. Sci. USA*, **99**, 13627–13632.
- Sonnhammer, E.L. and Durbin, R. (1997) Analysis of protein domain families in *C.elegans*. *Genomics*, **46**, 200–216.
- Vandepoele, K., Saeys, Y., Simillion, C., Raes, J. and Van De Peer, Y. (2002) The automatic detection of homologous regions (AD-HoRe) and its application to microcolinearity between *Arabidopsis* and rice. *Genome Res.*, **12**, 1792–1801.
- Vision, T.J., Brown, D.G. and Tanksley, S.D. (2000) The origins of genomic duplications in *Arabidopsis*. *Science*, **290**, 2114–2117.
- Vision, T.J. and Brown, D.G. (2000) Genome archaeology: detecting ancient polyploidy in contemporary genomes. In Sankoff, D. and Nadeau, J.H. (eds), *Comparative Genomics*. Kluwer, Dordrecht, pp. 479–492.
- Wolfe, K.H. (2001) Yesterday's polyploids and the mystery of diploidization. *Nature Genetics Reviews*, **2**, 333–341.