BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 8 2005, pages 1339–1348
doi:10.1093/bioinformatics/bti168

Genome analysis

Statistical detection of chromosomal homology using shared-gene density alone

S. E. Hampson[1,3], B. S. Gaut[2,3] and P. Baldi[1,3,*]

[1]School of Information and Computer Science, [2]Department of Ecology and Evolutionary Biology and [3]Institute for Genomics and Bioinformatics, University of California, Irvine, Irvine CA 92697, USA

Received on July 12, 2004; revised on November 4, 2004; accepted on November 18, 2004
Advance Access publication December 7, 2004

ABSTRACT

Motivation: Over evolutionary time, various processes including point mutations and insertions, deletions and inversions of variable sized segments progressively degrade the homology of duplicated chromosomal regions making identification of the homologous regions correspondingly difficult. Existing algorithms that attempt to detect homology are based on shared-gene density and colinearity and possibly also strand information.

Results: Here, we develop a new algorithm for the statistical detection of chromosomal homology, CloseUp, which uses shared-gene density alone to fully exploit the observation that relaxing colinearity requirements in general is beneficial for homology detection and at the same time optimizes computation time. CloseUp has two components: the identification of candidate homologous regions followed by their statistical evaluation using Monte Carlo methods and data randomization. Using both artificial and real data, we compared CloseUp with two existing programs (ADHoRe and LineUp) for chromosomal homology detection and found that in general CloseUp compares favorably.

Availability: CloseUp and supplementary information are available at http://www.igb.uci.edu/servers/cgss.html

Contact: pfbaldi@ics.uci.edu

1 INTRODUCTION

When a chromosomal segment begins to diverge, whether due to a speciation event or due to an intra-genomic duplication, the two homologous regions are initially identical in length and base sequence. Any genes or markers in the homologous regions are identical in order, spacing and strand. For a recently duplicated chromosomal segment, these properties are statistically extreme and the region is easily identified. However, over evolutionary time, point mutations, insertions, deletions and inversions progressively degrade these properties toward the statistical background, making precise identification of the homologous regions correspondingly difficult (Sankoff and El-Mabrouk, 2002; Durand and Sankoff, 2003; Calabrese *et al*., 2003). The relative rates of degradation of these properties is an open empirical question (Nadeau and Taylor, 1984; Schoen, 2000; Wolfe and Li, 2003), so it is not obvious which features, or combination of features, are most useful for detecting highly degraded homology. For example, if insertions and deletions dominate, the order and strand of shared genes/markers may be more conserved than their density. Conversely, if inversions dominate, density may be more informative. Algorithmic analysis of these properties can vary greatly in complexity, so the choice of features and methods of analysis can impact runtime as well as accuracy.

The detection of chromosomal homology is important both for identifying synteny between two genomes (Trachtulec and Forejt, 2001) and for identifying regions of intra-genomic duplication (Kent *et al*., 2003; Pevzner and Tesler, 2003a). Two practical algorithms for detecting chromosomal homology have been developed recently: the ADHoRe algorithm (Vandepoele *et al*., 2002), which uses density, order and strand information and the LineUp algorithm (Hampson *et al*., 2003), which uses density and order. It is important to note that while ADHoRe and LineUp emphasize order information, neither is completely dependent on it. For example, ADHoRe begins by considering order and strand information but can also extend candidate regions based purely on density information. Similarly, colinear run generation in LineUp permits a limited amount of local reordering to accommodate uncertainty in marker location or general degradation of order information. The larger the range of reordering permitted, the more relaxed the definition of colinearity and the greater the reliance on density over order.

A moderate amount of relaxation is probably always reasonable, but the reordering approach can be combinatorially expensive for high-density marker data, limiting the amount of relaxation possible. Two versions of LineUp were implemented to address this problem: one which attempts to generate all legal colinear permutations (FullPerm) and one which generates only a heuristic subset (FastRun). FastRun is more efficient in time and space and gives nearly identical homology results, but is still computationally intensive as a function of the degree of colinear relaxation.

The complexity of ADHoRe and LineUp, both in code and runtime, is determined primarily by the use of order information. Consequently, an important question is whether the ability to utilize order information is actually necessary for, or even significantly improves, homology detection under reasonable application conditions. In theory, the ability to use order information should only improve detection, but overly stringent order requirements might actually exclude homologous but jumbled regions (Sankoff and El-Mabrouk, 2002; Sankoff, 2002). Similarly, it is not entirely clear whether strand information is essential or useful.

Here we develop a new algorithm, CloseUp, both to exploit the fact that in our previous experience relaxed colinearity was observed to be generally beneficial for homology detection and to optimize

[*]To whom correspondence should be addressed.

computational time. CloseUp uses shared gene or marker density alone, without regard to strand or order (see the Methods section). For CloseUp, we have strived to retain the same implementation as LineUp as much as possible. This is not only because LineUp is already a successful homology detection program, but also to make sure that within essentially the same program, replacing the complex calculations of co-linearity with a much simpler calculation of density can actually lead to improved results. We compared ADHoRe, LineUp and CloseUp using both artificial and real data and found that in general CloseUp compares favorably with the other algorithms.

2 METHODS

The CloseUp algorithm can be viewed as a limiting case of the LineUp algorithm when its colinearity-relaxing parameters I, D and O all go to ∞ (Hampson *et al*., 2003). For completeness and clarity, it is better however to provide a self-contained description of CloseUp. CloseUp has two components: the identification of candidate homologous regions followed by their statistical evaluation using Monte Carlo methods and data randomization.

2.1 Identification of candidate regions

Candidate regions are identified by finding all homologous gene pairs between two chromosomes and attempting to extend each pair into two larger clusters. As seen in the results, CloseUp can compare a chromosome to itself but for the sake of exposition we will use two distinct chromosomes $C1$ and $C2$. The comparison is asymmetric, so the results of comparing $C1$ to $C2$ may not be the same as comparing $C2$ to $C1$. CloseUp automatically runs both comparisons. Comparison of $C1$ to $C2$ is achieved through an outer loop and an inner loop:

Outer loop: The outer loop cycles through all the genes on $C1$. For each gene on $C1$, it then cycles through all its matches on $C2$. Gene matches are precomputed and indexed, so no searching is necessary. The outer loop typically takes linear time in the sense that if N is the typical number of genes on a chromosome and $k \ll N$ is the maximal number of matches any gene on $C1$ has on $C2$, then the outer loop requires at most kN basic steps.

Inner loop: The inner loop progressively extends each gene pair, generated by the outer loop, into local clusters by using two simple tests of proximity and density, which are explained below, to control the extension process. In the inner loop, we assume that a gene A on $C1$ has been matched to a gene A' on $C2$. We initialize the corresponding developing clusters as $\mathcal{C}_1 = \{A\}$ on $C1$ and $\mathcal{C}_2 = \{A'\}$ on $C2$. Then:

(1) Move to the next gene X to the right of $\mathcal{C}_1$. Set $\mathcal{C}_1 = \mathcal{C}_1 \cup \{X\}$.

(2) Find the gene X' homologous to X on $C2$ that is closest to $\mathcal{C}_2$ (on its right or its left). If no gene is found, go to Step 1.

(3) Test X' for proximity to $\mathcal{C}_2$, by comparing the length of $\mathcal{C}_1$ and $\mathcal{C}_2 = \mathcal{C}_2 \cup \{X'\}$. If the proximity test fails, do not include X in $\mathcal{C}_2$ and go to Step 1 to continue the expansion.

(4) Set $\mathcal{C}_2 = \mathcal{C}_2 \cup \{X'\}$, where $\{X'\}$ represents X' and all the genes between $\mathcal{C}_2$ and X'.

(5) Test $\mathcal{C}_1$ and $\mathcal{C}_2$ for density. If dense, save $\mathcal{C}_1$ and $\mathcal{C}_2$ and go to Step 1, else stop.

The inner loop saves all possible cluster pairs containing three or more shared genes. As in LineUp, subclusters and overlapping clusters are retained for statistical evaluation by Monte Carlo methods. Retaining clusters of all size provides greater robustness and sensitivity since large clusters are made of smaller subclusters and portions of a cluster may be more significant than the entire cluster.

The number of steps in the inner loop is commensurate with the size of the clusters. Thus if the clusters are bound in size by l, the complexity of the entire algorithm scales like $O(klN)$, linear in the chromosome length. This scaling is similar to LineUp, except for one important element. As the order constraint is relaxed, LineUp can search up to all viable permutations within a run (we use 'run' in LineUp instead of 'cluster' in CloseUp when order matters), and therefore the inner loop in LineUp can scale like $l!$, which rapidly becomes prohibitively slow even for reasonable values of l.

Expanding the region on $C1$ to the right (Step 1) introduces some asymmetry. However, right-to-left or bidirectional expansion do not greatly change the results. Thus, for consistency and comparison with LineUp they are not used in the results presented here. Bidirectional expansion is provided, however, as a runtime option of CloseUp.

In a similar vein, genes with no corresponding matches between $C1$ and $C2$ have no impact in LineUp, but have a generally small effect in CloseUp since they affect the density calculation. An option to retain or exclude unmatched genes is provided and, for consistency and comparison with LineUp, the exclude option is used in the examples presented here.

Additional changes might further improve the results, but would obscure the important point that, within essentially the same program, replacing the complex calculations of colinearity with a much simpler calculation of density can actually lead to improved results.

2.1.1 Test of proximity Note that the proximity test in Step 3 of the inner loop does not terminate cluster expansion. If X' is too far, it is merely excluded from $\mathcal{C}_2$ and expansion continues with Step 1. The proximity test is performed by checking whether the absolute value of the difference in lengths of $\mathcal{C}_1$ and $\mathcal{C}_2$ is < 20 times the pre-computed average inter-distance between genes in the chromosomes. This test is implicitly based on the idea that, to a first degree of approximation, homologous regions ought to have roughly the same length and number of genes. For example, consider the situation:

```
C1: ABTCDxxxxxxxxxxxxxxxxxxxxxxxxxxxS
C2: ABSCDxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxT
```

This situation might result, for example, from four random insertions of the transposons S and T. When comparing $C1$ to $C2$, if the matching Ts are included in the nascent AB cluster, the resulting $C2$ region would be grossly over-extended beyond the region of true homology, and similarly for the S matches when comparing $C2$ to $C1$.

The general problem is that there are matches due to homology which can be used to define homologous regions, and matches due to other processes, such as transposons, which have to be filtered out. The actual cluster (ABCD) can easily be obscured by random background matches (S and T). LineUp can address this problem by locally re-ordering the genes, to recover the colinear run ABCD. CloseUp does not use the computationally expensive local re-ordering mechanism, and simply skips any match on $C2$ that makes the difference in length between the two regions > 20 times the average gene distance. This is similar to the 'skip' mechanism (I parameter) used in LineUp (Hampson *et al*., 2003) which permits some matching markers to be ignored. Values between 10 and 40 gave similar results on a number of real and artificial datasets (see the Results section), so no further attempt was made to 'tune' this parameter to individual datasets. Further refinement and automatic adjustment of this parameter are topics of future research.

2.1.2 Test of Density In algorithms that rely on order information, candidate homologous regions are terminated when colinear extension is no longer possible. For the identification of cluster pairs in CloseUp, order-based termination is replaced with density-based termination. Here, we provide a method for identifying relatively high-density cluster pairs based on the expected number of shared genes between two random sets of genes. Region extension is terminated if the observed number of shared genes is not significantly greater than expected, in practice, twice.

To derive a rough estimate for the expected number of matches between a set of n genes on $C1$ and a set of m genes on $C2$, imagine that the two chromosomes contain a total of $2N$ genes subdivided into K classes of non-homologous genes. Then for large N and K and randomly distributed genes, the number of matches approximately follows a binomial distribution with nm trials and success probability equal to $1/K$. As a result, the expected number of matches is approximately equal to nm/K

with variance $\sigma^2 = nm[1/K][(K - 1)/K]$. Thus deviations from the expected value can be measured using fold differences or, more rigorously, Z-scores that take into account the variance. Both options are available in CloseUp.

2.2 Evaluation of candidate regions

Cluster pair detection, and all its variations, provides an initial filter for what might be homologous regions. The proximity and density tests, however, are not used as statistical tests but rather as filtering procedures to limit the number of cluster pairs that need to be considered. The resulting filter is sufficiently broad that many cluster pairs will occur purely by chance. Consequently, it is necessary to identify the cluster pairs that are least likely to occur by chance. As in LineUp (and ADHoRe), statistical evaluation is carried out using Monte Carlo methods, by randomizing the data many times and ranking cluster pairs highly if they have an unusually large number of shared genes in an unusually short physical region relative to randomized data. Real homologous regions are thus assumed to be dense.

Two properties are used to evaluate a cluster pair: their size (number of matched genes) and their length on the chromosome (e.g. in cM or bp). Various measures of length have been tried previously (Gaut, 2001; Hampson *et al*., 2003), but all gave similar results. We use the Summed Squares (SS) metric here. The SS value of a cluster pair is the sum of the squared lengths of each cluster.

To test for statistical significance for each chromosome pair ($C1$, $C2$), gene identifiers are permuted on $C2$ and cluster detection repeated. This is done for a large number of times (generally 1000) and the resulting cluster pairs are binned by the number of matched genes. This provides an estimate of the background frequency and distribution of lengths for the cluster pairs in each bin.

There are several subtleties and alternative approaches for choosing a decision boundary between significant and non-significant clusters, across cluster sizes and lengths, which are further discussed in the Appendix section. Two important points to keep in mind are that: (1) candidate clusters identified by the above procedure are not independent since subclusters and overlapping clusters are explicitly retained and (2) as the size of the clusters increases, the number of randomized data clusters in the corresponding bin diminishes and therefore at some point it becomes difficult to get an accurate estimate of the length distribution for clusters of a certain size. In addition, as observed previously for LineUp (Hampson *et al*., 2003), CloseUp also runs a significant risk of cluster over-extension. Specifically, once a long real cluster pair has been discovered, any legal extension of it will automatically be considered to be significant since there are no randomized cluster pairs of that size to compare it to. It was observed that limiting the maximum run length in LineUp to a range (typically 10) where statistical comparison is possible largely eliminates the problem and also reduces memory and time requirements.

Thus to address these issues in the simulations we retain only clusters of size up to L (typically $L = 10$), where L corresponds to a bin size containing a reasonably large number of clusters derived by the random permutation procedure, i.e. a number of clusters larger than the number of permutations (generally 1000). Accordingly, in the following simulations cluster or run size is limited to 10 for both LineUp and CloseUp. With a significance threshold percentage of $x/100$ (typically 1%), we then select the $x/100$ fraction of clusters of size $3, 4, \ldots,$ and L with the shortest length. That is for each candidate cluster pair, the number of random cluster pairs with the same gene number but smaller SS value is tallied. This is divided by the number of cluster pairs in the bin and thresholded at $x/100$. We generally use a 1% significance threshold, but results can be tuned according to the empirical distribution of probability values. Note that, as for LineUp, large clusters of size greater than L are not lost and can still be recovered as unions of smaller subclusters.

Three methods of randomizing the markers on $C2$ were tested: (1) assigning random values between the biggest and smallest observed in the real data on that chromosome; (2) leaving the locations the same but permuting the marker names; and (3) leaving the locations the same but permuting the marker names only for those markers that matched between the two chromosomes. The three methods provided qualitatively similar results (data not shown) and we report analyses based on the second method.

One standard way to characterize the quality of detection algorithms is by using receiver operating characteristic (ROC) curves. ROC curves are used in statistics and signal detection to study the tradeoff between false positives and false negatives for different detection cut-off values. More precisely, an ROC curve is a plot of the hit rate (or sensitivity) defined by TP/(TP + FN) versus the false alarm rate FP/(FP + TN) at each possible threshold, where TN, TP, FN and FP are true negatives, true positives, false negatives and false positives, respectively. In the simulations, we used ROC curves to compare algorithms. To draw exact ROC curves and/or to compare different algorithms; however, the 'correct' answer must be known and therefore this is best done on artificial data. Artificial data allows us also to deal with the problem of defining FP and FN in the case of partial clusters by looking at the projection of significant clusters onto the region of known homology, as described in the Results section. Approximate ROC curves can be inferred from distributions of P-values using the methods described by Baldi and Hatfield (2002) and Hung *et al*. (2002), and modeling these distributions as mixtures of beta distributions, with one of the components being uniform and corresponding to the case where the null hypothesis is true.

CloseUp is available for download at http://www.igb.uci.edu/servers/cgss.html

3 DATA

We used both real and simulated data to compare CloseUp, LineUp and ADHoRe.

3.1 Maize marker data

To compare the performance of CloseUp and LineUp on real data, we used the Pioneer 1999 Composite map (Pio99) of the maize genome, which is an amalgamation of several maize molecular genetic maps. The Pio99 map contains 2415 markers that hybridize to more than one location on the ten chromosomes of the maize genome. Thus, Pio99 data constitutes a reasonable dataset for identifying homologous regions between chromosomes within the maize genome. Pio99 data were downloaded from http://www.agron.missouri.edu/, and LineUp results on Pio99 have been reported previously (Hampson *et al*., 2003).

3.2 Simulated data

In addition, for the purpose of algorithm testing and comparison, we also use simulated data. Simulated data present at least two advantages for assessing algorithmic performance and accuracy: (1) the difficulty of the problem, in terms of degraded homology, can be adjusted in a controlled fashion; (2) the 'correct' answer is known in advance, so that error rates and ROC curves can be determined precisely. Artificial data must be constructed using biologically realistic parameters in order to provide a meaningful point of comparison.

Three main parameters were used to generate artificial data: (1) background similarity: what fraction of genes not located in the homologous region are similar to each other? (2) Conserved genes: what fraction of the duplicated genes are still present in homologous regions? (3) Conserved order: to what extent is the initial order of the genes retained in the homologous region? The first parameter—background similarity—will vary depending on the organism and the type of data (markers in a molecular genetic map versus genes from a genome sequence). In general, the second and third parameters will approach the background given sufficient evolutionary time for deletions, rearrangments and other degradation events to occur.

Artificial data are generated in three steps corresponding to the three parameters above. In general, we have used sequence characteristics from *Arabidopsis thaliana* (*Arabidopsis* Genome Initiative, 2000) as a guideline for simulated data.

First, a uniform background is created. This is done by assigning $2N$ unique genes to random locations and random strands on two chromosomes, $C1$ and $C2$, N genes per chromosome. For *Arabidopsis*, average chromosome length is ∼25 million bases (*Arabidopsis* Genome Initiative, 2000), and that value is used here, although the actual value used is largely immaterial. For *Arabidopsis*, N is in the order of 5000, but for simulation efficiency $N = 1000$ is used here. The N genes are placed randomly along the length of the chromosome. To create a background similarity distribution, each of the $2N$ genes is considered and with probability R is replaced with another gene chosen at random from the $2N$ set. This produces a background distribution of singles (found only once on $C1$ and $C2$), doubles (found twice on $C1$ and $C2$), triples, and so forth, depending on the value of R. Typically we used a value of $R = 0.2$, which is again similar to *A.thaliana* (*Arabidopsis* Genome Initiative, 2000; Vision *et al*., 2000). The larger the value of R, the harder the problem for at least two reasons. First, there is simply a larger background of potential, but random, homology, and separating signal from noise becomes more difficult as the noise increases. Second, the additional background pairs may break up the real homology region as in the above transposon example (Section 2.1.1).

Next, a region of homology comprising the middle 20% of each chromosome is created. For $N = 1000$, the region contains 200 genes, which can be numbered in any order. A random fraction S of the genes in that region on $C2$ are replaced with the correspondingly numbered genes from that region on $C1$. Strand and location information are also duplicated, so if $S = 1$ the two regions are identical. When $S < 1$, the number of unpaired genes in the homologous regions reflects the amount of insertion/deletion that occurred after duplication. In our simulations, $S = 0.1$, 0.2 and 0.3, again approximating the *A.thaliana* genome, for which ∼28% of genes are retained in duplicated regions (Blanc *et al*., 2003; Simillion *et al*., 2002).

Finally, F randomly chosen neighboring gene pairs are inverted in the homologous region of $C2$. That is, only small-scale inversions are simulated. With increasing F, strand, order and local density are progressively degraded. If inversions are performed only within the homologous region, the boundaries remain distinct, but if inversions are permitted over the entire chromosome, the homologous and non-homologous regions 'diffuse' into each other. In that case the actual extent of the homologous region becomes subjective, so for testing purposes we limited the inversions to the boundaries of the homologous region. Consequently, the density of shared genes over the entire region does not change while the density of shared genes in smaller subregions varies. F was varied between 0 and 10 million, which is sufficient to completely randomize gene order in the homologous region. The upper limit of 10 million inversions was empirically determined, as follows. An array V of length v was initialized to $V[i] = i$. Initially, $\sum_i |V[i] - i|$ is zero, but with successive neighbor inversions, the sum increases to the long-term average of $v^2/3$. For $v = 200$, this value is reached between 1 and 10 million inversions.

4 RESULTS

LineUp, CloseUp and ADHoRe were used primarily with default parameter settings. The reordering parameter, D, for LineUp was set to twice the average distance between markers; that is, markers separated by less than D could be reordered to achieve colinearity. CloseUp was applied with density and proximity values of 2 and 20 as discussed previously. Initial ADHoRe results gave relatively poor results, so a number of parameter adjustments were attempted (data not shown). All were ineffective except increasing the ADHoRe.pl 'max-dist' value (Vandepoele *et al*., 2002), which increased detection of the homologous region but also increased the proportion of the non-homologous region that was falsely identified as homologous. Consequently, this value was increased from the suggested 20 to 40, which yielded maximum homologous coverage without producing greater non-homologous coverage than the other algorithms.

4.1 Comparison on the maize Pio99 dataset

Despite their different objective functions, LineUp and CloseUp yield similar results when applied to the Pio99 dataset. ADHoRe could not be applied to this dataset since strand information is not available. The thresholds of the two algorithms were adjusted to yield approximately the same number of homologous regions. Unlike artificial data where asymptotic positive coverage is achieved with a threshold of 1%, the results with the Pio99 data are not stable until well past 5% (Fig. 4), further stressing the importance of statistical methods, such as ROC curves.

Visual inspection of the results in Figures 1 (CloseUp) and 2 (LineUp) indicates at least 80% correspondence between the identified regions. If the thresholds are relaxed slightly, well over 90% of the regions in Figures 1 and 2 are matched, indicating that any disagreement between the two methods is more a matter of ordering the positive instances than detecting different sets. This implies that high colinearity and high density are correlated. We take the correspondence in results both as evidence that the common elements are real and that density information alone can identify homologous regions. Note, however, that while there is a good agreement between the identified regions, the boundaries often differ, again indicating the difficulty of defining the true extent of regions of highly degraded homology.

It is worth noting that, despite their high-statistical significance, some of the best runs and cluster pairs found in the Pio99 dataset are probably spurious. For example, in Figures 1 and 2, note that the same location of chromosome 3 pairs significantly with many other chromosomes (e.g. chromosomes 0, 1, 3, 4, 5, 6, 7, 8 and 9), using the 0–9 numbering scheme of these figures rather than the usual 1–10 scheme. These results seem to stem from the fact that many markers on chromosome 3 map to the same position. This could be biologically real, but more likely reflects a lack of resolution in the map. Similar results can also be found with chromosomes 2 and 5.

4.2 Comparison on simulated data

All three algorithms were compared on artificial data. As noted, simulated data were generated with $R = 0.2$, S-values of 0.1, 0.2 and 0.3, and F-values between 0 and 10 million (complete order randomization). For all three methods, 100 cycles of dataset randomization were used for each test. Each test was replicated 100 times and the average percentage coverage of homologous and non-homologous regions reported. All algorithms were applied with a significance threshold of 0.1%.

```
0: ...................111.................................................
0: .......222...............................22.....22.....................
0: ...........................................33..........................
0: ............................................44444444444444444444.......
0: .......................................66666666666.....................
0: ...................77...............77777777777777.....................
0: ..............88888888.................................................
1: ............................000........................................
1: .......................11.....1........................................
1: ......................3333333333.......................................
1: ....................55555..............................................
1: ..........................................66666...6666.................
1: ....99999999999999.....................................................
2: .............000.........0.........00..................................
2: .............111.......................................................
2: .........................2.............................................
2: .........................333...........................................
2: .............................................................44........
2: .....................7777777777..777777.......77777......7777777.......
2: .........................888...........................................
2: .........................99999999999...................................
3: ........................000000.........................................
3: ........................1..............................................
3: ........3.....3333333333333............................................
3: ......................44444444...........4444444.......................
3: ........................555............................................
3: .........666............666666..............................6666.......
3: ........................777............................................
3: ........................8..............................................
3: .........................99............................................
4: ....0000000000000000000000.............................................
4: ..................................1111.................................
4: ..................................2....................................
4: ...................................3.3333333....333...3333.............
4: .................................77777777777...........................
5: .....................................................000000............
5: ..........222..........................................................
5: ............5..........................................................
5: ..........777......................777777777777777777..................
6: ............1111111111.................11111111111111111111............
6: .....................3.................................................
6: ...............................................55555...................
7: ...............................00000...................................
7: ..2222............2222222..........2222222.222.222222222...............
7: ................................5555...................................
7: ......................888888888888888..................................
7: ......................9................................................
8: .............................0000000000000000000000.....000............
8: ...........................222........2222.............................
8: ...............................777.....................................
8: .............................8888......................8888............
9: .................................................000...................
9: ..................................11.................11111111111.......
9: .....................2222222222222.....................................
9: ...77..................................................................
```

Fig. 1. CloseUp results on Pio99 maize marker dataset. Each chromosome ($C1 = 0$–9) aligned against all chromosomes ($C2 = 0$–9). Maize chromosomes 1–10 are numbered 0–9 instead for ease of presentation. For each $C1$–$C2$ pair, regions of homology are shown in each row using the number of $C1$ followed by a representation of the chromosome with the regions of homology marked with the numeric value of $C2$. Therefore, for example, chromosome 0 has three short homologous regions with chromosome 2 and one long homologous region with chromosome 4. Probability cut-offs for the two algorithms adjusted to give about the same total number of homologous regions.

```
0: ...................111.................................................
0: .......222..........22...................22.....22.....................
0: ...................................33......33..........................
0: .............................................444444444444444444........
0: ................888888.................................................
1: ............................000........................................
1: ............................111........................................
1: ......................33333333.........................................
1: ..........................................444..........................
1: ....................555................................................
1: ............................................66.....66666...............
1: ....................88888888888........................................
2: .............000000.000000.........00.........................00.......
2: .........................2.............................................
2: .........................333...........................................
2: .............................................................44........
2: .........................777........777777......77.......7777777.......
2: .........................888...........................................
2: .........................9...99........................................
3: ....................000000000000................0......................
3: ........................11.................111.........................
3: .......3333.......333...333............................................
3: ......................4444...444444444444444444.4444444......4444444444
3: .........666.......666666666...........................................
3: ........................777............................................
3: ........................8..............................................
4: ....0000000000000000000000.............................................
4: ......................111..............................................
4: ..................................2....................................
4: ...................................333.333......333...333..............
4: .....................................7777777...........................
5: ............000......................................000000............
5: ..........1............................................................
5: ..........222222222....................................................
5: .......333333..........................................................
5: ..........5.5..........................5...............................
5: ..........777..........................77..77..7.......................
6: ..................1111.................11111111111111111111............
6: .....................3.................................................
7: ..2222.............222222..........222.......22222222222...............
7: ......................3................................................
7: ........................555........5...................................
7: ..........................666..........................................
7: ............................7..........................................
7: .....................................................888...............
7: ......................99...............................................
8: .......................000000....0000.000000000000......000............
8: .............................111111111.................................
8: ......................................2222.............................
8: ................33.....................................................
8: .........555555555.....................................................
8: ................................77.....................................
8: ...........................888..................88.....88..............
9: ......................................111111.........1111..11111.......
9: ..........................22222222.....................................
9: ...7...................................................................
```

Fig. 2. LineUp results on Pio99 maize marker dataset. Each chromosome ($C1 = 0$–9) aligned against all chromosomes ($C2 = 0$–9). Maize chromosomes 1–10 are numbered 0–9 instead for ease of presentation. For each $C1$–$C2$ pair, regions of homology are shown in each row using the number of $C1$ followed by a representation of the chromosome with the regions of homology marked with the numeric value of $C2$. Therefore, for example, chromosome 0 has four short homologous regions with chromosome 2 and one long homologous region with chromosome 4. Probability cut-offs for the two algorithms adjusted to give about the same total number of homologous regions.

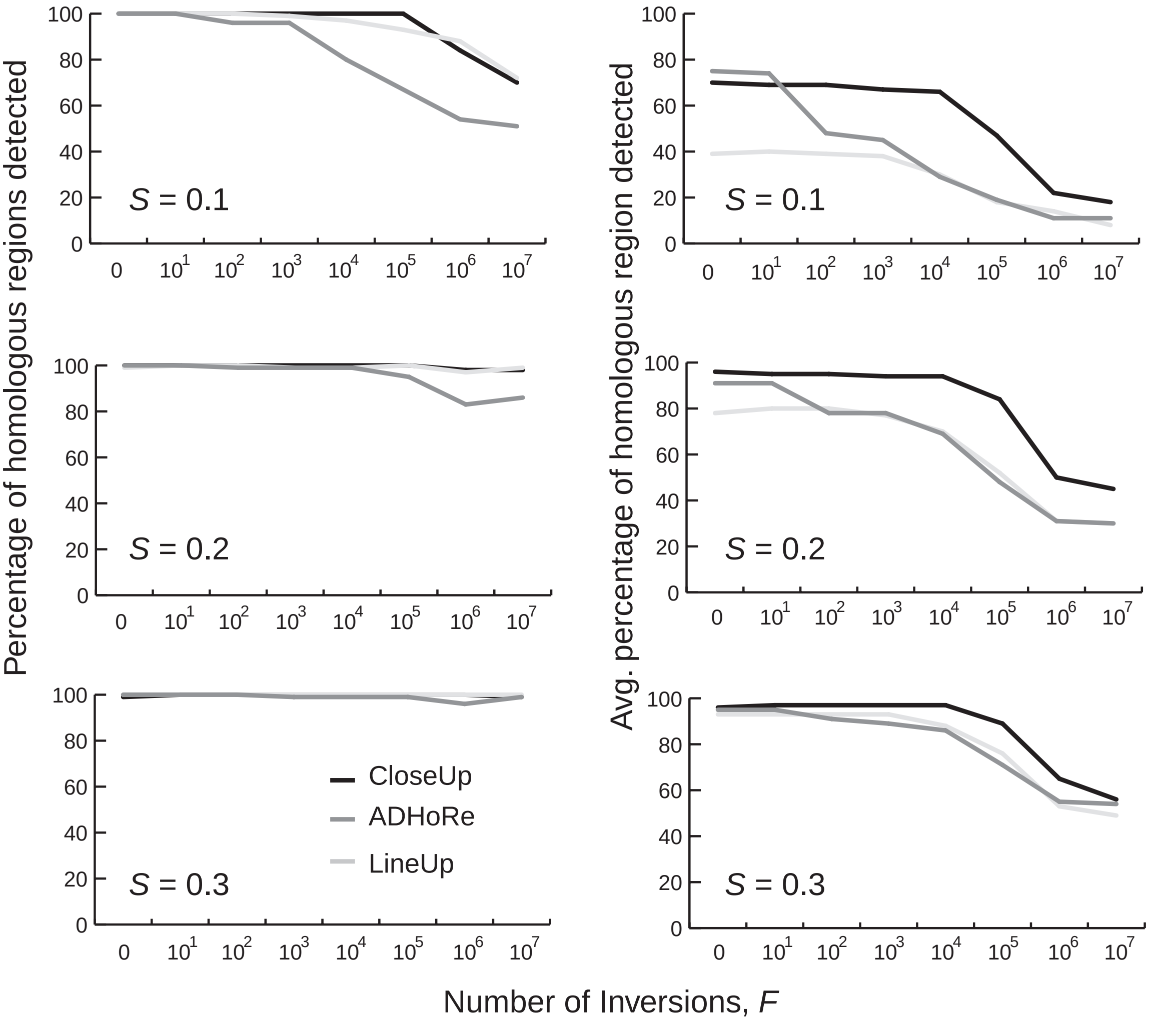

Fig. 3. Comparison of CloseUp, LineUp and ADHoRe on artificial datasets. Conserved gene fraction S values of 0.1, 0.2 and 0.3, and number F of inversions between 0 and 10 million. Proportion of datasets in which a method identified some portion of the homologous region and the average proportion of the homologous region that was properly identified.

Figure 3 graphs two important aspects of the results: the proportion of datasets in which a method identified some portion of the homologous region and the average proportion of the homologous region that was properly identified. There are some clear trends with all three algorithms. For example, the identified proportion of the homologous region decreases with decreasing gene conservation (decreasing S) and decreasing order conservation (increasing F). Overall, LineUp and ADHoRe give comparable results, although the limitations of LineUp's reordering mechanism are apparent when genes of the homologous region are widely separated (low S). In this case, LineUp identifies a considerably smaller proportion of the homologous region. CloseUp compares favorably at all tested conditions, although as expected it is weakest when density information is more degraded than order information (low S and low F).

Previous studies have shown that homology detection algorithms can also falsely identify non-homologous regions (Hampson *et al.*, 2003). Although some non-homologous regions were incorrectly identified in simulated data, in no case did any of the three algorithms falsely identify >3% of the non-homologous region (data not shown). The false identification of non-homologous coverage is largely, although not entirely, due to over-extension beyond the boundaries of the homologous region.

Analysis of simulated data is much faster with CloseUp than LineUp, which is somewhat faster than ADHoRe. On a single Pentium processor CloseUp took a few hours to run the test set, LineUp ∼2 days and ADHoRe ∼4 days. Under other circumstances (e.g. LineUp with a larger D value that permits more marker reordering) ADHoRe can be faster than LineUp, but CloseUp is always much faster than the other two. Within reasonable regimes, both CloseUp and LineUp (Hampson *et al.*, 2003) have linear time complexity in the number of genes or markers, but it is the slope of the line that can differ significantly.

These results should not be taken as a comprehensive quantitative comparison of the different approaches. All three algorithms have parameters that could be optimized for individual test conditions, quite possibly improving performance noticeably. For example,

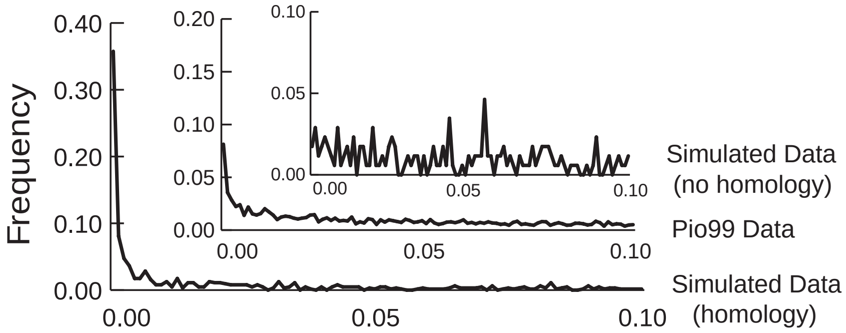

Fig. 4. Frequency of cluster pairs binned in 0.1% significance steps (Section 2.2) from 0 to 10% for real data (Pio99) and simulated data with and without homology. Peaks in the distributions for significance values near 0 correspond to homologous clusters.

a fixed significance cut-off of 0.1% was used in all tests. This choice was based on visual inspection of the number of significant runs/cluster pairs at different cut-off values, but there is nothing intrinsically special about it. Figure 4 shows the frequency of cluster pairs found by CloseUp versus the significance cut-off when binned in 0.1% increments between 0 and 10%. Three datasets are shown: real data (Pio99) and a representative example of artificial data ($S = 0.2, F = 10^4$) with and without a homologous region. Data without a homologous region demonstrate a uniform distribution, but the other two show a clear peak for significance values close to 0, presumably representing true homologous cluster pairs. Comparison of these distributions provides insight into the best threshold for a given dataset. For artificial data, the peak is quite narrow and a threshold of 1% gives good coverage with a low false positive rate, while for real data, a somewhat larger threshold is required for similar coverage, but with a correspondingly higher false positive rate. All of the curves are flat (uniform distribution) beyond 10%.

ROC curves allow one to further analyze this point by formalizing the tradeoff between sensitivity and specificity. In Figure 5, empirical ROC curves for LineUp and CloseUp are shown for a representative example ($S = 0.2, F = 10^4$), averaged over 1000 tests. To generate these curves, each chromosome was divided into 100 equal-sized regions and the set of significant runs/clusters projected onto it. The significance cut-off was incrementally relaxed until near asymptotic coverage was achieved, providing a set of instances that can be analyzed in terms of TP, TN, FP and FN. With artificial data, TP, TN, FP and FN values do not have to be estimated since the correct P (homologous) and N (non-homologous) sets are known. The resulting curves clearly indicate that CloseUp is uniformly superior to LineUp in this example since, for any threshold or any FP rate, it has a higher proportion of TP.

ROC curves could be determined for CloseUp and LineUp in this way, but ROC curves could not be produced for ADHoRe because adjusting the final significance cut-off in ADHoRe had little effect on results. Presumably this is because most of the filtering in ADHoRe is done in various ways during the generation phase, so changing the final cut-off in the evaluation phase often has little effect. Tinkering with the filtering steps in the program did not seem suitable or appropriate for external users. Thus ADHoRe results are summarized by isolated points rather than complete curves. For the particular artificial test for which we have ROC curves, the best ADHoRe ROC point is ($x = 0.0, y = 0.53$). Relaxing the cut-off has no effect, while tightening it leads to worse performance ($x = 0.0, y = 0.38$).

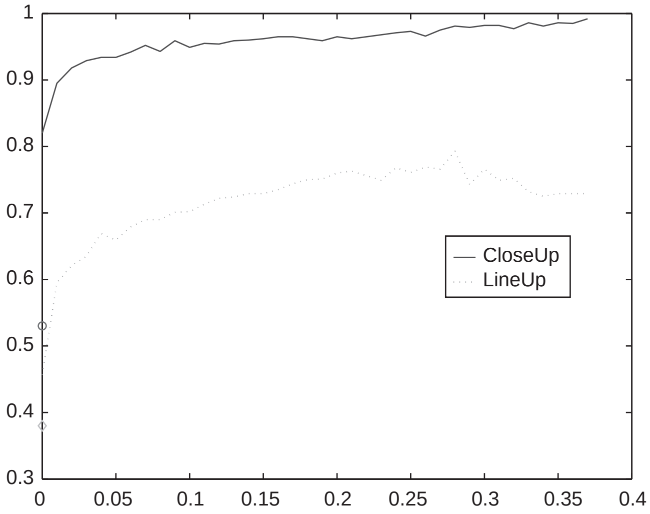

Fig. 5. ROC curve comparing CloseUp, LineUp and Adhore on the artificial dataset. Each curve is the average of 1000 tests. $x = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})$ and $y = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$. Single points circle $= (0.00, 0.53)$ and diamond $= (0.00, 0.38)$ correspond to ADHoRe.

5 CONCLUSION

We have developed a new algorithm, CloseUp, for the detection of chromosomal homology. Although motivated by our previous algorithm—LineUp (Hampson *et al.*, 2003)—CloseUp differs substantially in that it depends on density information as opposed to order information. This approach is motivated by the observation that density appears to be a more durable measure of homology than order. For instance, inversions that preserve order significantly will also preserve density, although the converse is not true. A series of nested or overlapping inversions, for instance, would progressively disrupt order and small-scale density, but would not affect large-scale density.

In LineUp, the region on $C2$ is extended to the right, or on a second separate iteration to the left, as long as colinearity with the region on $C1$ is maintained. In CloseUp only a single iteration is made and matching genes on $C2$ are chosen to be as close as possible to the developing cluster. The goal is to determine if genes in the region on $C1$ are also clustered on $C2$. As in LineUp, subregions and overlapping regions are retained for statistical evaluation.

CloseUp should be regarded as a fast algorithm for detecting putative homologous clusters or synteny blocks between chromosomes. Other algorithms (Pevzner and Tesler, 2003a,b) can then be applied to further study the homology between the blocks and analyze other properties, such as the minimal number of local inversions required to transform one block into the other. Future work ought to focus on the detection of density effects at multiple different scales beyond what can be captured by the current version of CloseUp.

One important qualitative conclusion can be drawn: for both real and artificial data, density information alone can be utilized efficiently and productively to identify homologous regions. In future work we hope to further improve the accuracy of the CloseUp method without sacrificing its simplicity. Comparison of LineUp and CloseUp on real data should help to identify strengths and weaknesses of the two approaches and more realistic artificial test generation should be possible as more biological data and analyses become available. While order and strand information is clearly not necessary for good performance, CloseUp could be extended to include some form of easily computed order information, such as the fraction of pairwise gene inversions in $C2$. This fraction goes from 0 (perfect colinearity) to 1 (perfect reverse colinearity). Not surprisingly, this value is low when F is low and approaches 0.5 when F is high, and many cluster pairs in Pio99 show good colinearity by this measure. In general, decreasing shared-gene density is correlated with decreasing colinearity. If useful, this simple measure could be included as an additional factor in the evaluation of pairs of clusters.

It appears that order and strand information are degraded prior to density information over evolutionary time (Seoighe *et al*., 2000; Huynen *et al*., 2001; Eichler and Sankoff, 2003). If this is true, strand and order information, while always potentially useful, are generally unnecessary for detecting high-quality homology and are largely uninformative for detecting highly degraded homology. Owing to this, the significantly increased code complexity and runtime incurred to deal with those properties may not produce significantly better results. In addition, too stringent an order requirement is apt to miss regions of highly degraded homology for which density but little order information remains. Consequently, rather than defining runs based on colinearity and then filtering them for density as LineUp does, it may be better to initially focus on density and only sparingly (if at all) utilize strand or order information.

ACKNOWLEDGEMENTS

We would like to thank Yimeng Dou for help with the CloseUp website. The work of P.B. is supported by grants from the NIH (LM-07443-01), NSF (EIA-0321390), Sun Microsystems and a Laurel Wilkening Faculty Innovation Award at UCI. The work of B.G. is supported by the NSF.

REFERENCES

Baldi,P. and Wesley Hatfield,G. (2002) *Microarrays and Gene Expression. From Experiments to Data Analysis and Modeling*. Cambridge University Press, Cambridge.

Blanc,G., Hokamp,K. and Wolfe,K.H. (2003) A recent polyploidy superimposed on older large-scale duplications in the arabidopsis genome. *Genome Res.*, **13**, 136–144.

Calabrese,P.P., Chakravarty,S. and Vision,T.J. (2003) Fast identification and statistical evaluation of segmental homologies in comparative maps. *Bioinformatics*, **19** (Suppl. 1),74–80.

Durand,D. and Sankoff,D. (2003) Tests for gene clustering. *J. Comput. Biol.*, **10**, 453–482.

Eichler,E.E. and Sankoff,D. (2003) Structural dynamics of eukaryotic chromosome evolution. *Science*, **301**, 793–797.

Gaut,B.S. (2001) Patterns of chromosomal duplication in maize and their implications for comparative maps of the grasses. *Genome Res.*, **11**, 55–66.

Hampson,S., McLysaght,A., Gaut,B. and Baldi,P. (2003) Lineup: statistical detection of chromosomal homology with applications to plant comparative genomics. *Genome Res.*, **13**, 999–1010.

Hung,S., Baldi,P., Wesley Hatfield,G., Hung,S. and Baldi,P. (2002) Global gene expression profiling in *Escherichia coli* K12: the effects of leucine-responsive regulatory protein. *J. Biol. Chem.*, **277**, 40309–40323.

Huynen,M.A., Snel,B. and Bork,P. (2000) Inversions and the dynamics of eukaryotic gene order. *Trends Genet.*, **17**, 304–306.

Arabidopsis Genome Initiative (2002) Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana. Nature*, **408**, 796–815.

Kent,W.J., Baertsch,R., Hinrichs,A., Miller,W. and Haussler,D. (2003) Evolution's cauldron: duplication, deletion and rearrangement in the mouse and human genomes. *Proc. Natl Acad. Sci. USA*, **100**, 11484–11489.

Nadeau,J.H. and Taylor,B.A. (1984) Lengths of chromosomal segments conserved since divergence of mouse and man. *Proc. Natl Acad. Sci. USA*, **81**, 814–818.

Pevzner,P. and Tesler,G. (2003a) Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. *Genome Res.*, **13**, 37–45.

Pevzner,P. and Tesler,G. (2003b) Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolution. *Proc. Natl Acad. Sci. USA*, **100**, 7672–7677.

Sankoff,D. and El-Mabrouk,N. (2002) Genome rearrangements. In *Current Topics in Computational Biology*. MIT Press, New Haven, CT.

Sankoff,D. (2002) Short inversions and conserved gene clusters. In *Proceedings of the 2002 ACM symposium on Applied Computing*, ACM Press, pp. 164–167. NY.

Schoen,D.J. (2000) Comparative genomics, marker density and statistical analysis of chromosome rearrangements. *Genetics*, **154**, 943–952.

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) A recent polyploidy superimposed on older large-scale duplications in the *Arabidopsis* genome. *Proc. Natl Acad. Sci. USA*, **99**, 13627–13632.

Trachtulec,Z. and Forejt,J. (2001) Synteny of orthologous genes conserved in mammals, snake, fly, nematode and fission yeast. *Mamm. Genome*, **3**, 227–231.

Vandepoele,K., Saeys,Y., Simillion,C., Raes,J. and Van De Peer,Y. (2002) The automatic detection of homologous regions (ADHoREe) and its application to microcolinearity between arabidopsis and rice. *Genome Res.*, **11**, 1792–1801.

Vision,T.J., Brown,D.G. and Tanksley,S.D. (2003) The origins of genomic duplication in the *Arabidopsis* genome. *Science*, **290**, 2114–2117.

Wolfe,K.H. and Li,W-H. (2003) Molecular evolution meets the genomics revolution. *Nat. Genet.*, **33**, 255–265.

APPENDIX: SIGNIFICANT CLUSTERS

The Monte Carlo procedure produces a distribution of cluster pairs across their size (number of shared genes) and lengths. Many different strategies can be used to define a significance decision boundary. One possibility is to reduce the problem to one dimension by defining a notion of density, essentially dividing size by length and then applying a standard percentage cut-off. However, this is unlikely to result in an optimal strategy because information is lost by collapsing size and length into a single number. If we keep a two-dimensional approach, then we can define several possible decision rules, and these may or may not return a fixed pre-determined percentage of the total number of clusters under consideration.

One key aspect is how to deal with large clusters. If we want the decision rule to return $x/100$ of all the clusters, one possible approach is to take $x/100$ of the clusters at each size. The corresponding length cut-off, however, becomes problematic for large sizes associated with low counts in the Monte Carlo procedure. With few observed clusters, it is difficult to estimate the corresponding distribution of sizes and determine the $x/100$ cut-off value. One possibility is to assume that the distribution is roughly Gaussian and estimate the $x/100$ cut-off value from the estimates of the mean and standard deviation. This approach based on moments, however, does not yield very accurate results in general. Another possible strategy is to accept all the large clusters of size greater than some threshold L. If this produces a fraction $y/100$ with $(y < x)$ of all possible clusters, then we can take a fraction $(x - y)/100$ of clusters at each length $1, 2, \ldots, L$ and recover a total number of clusters still equal to $x/100$. Here, we have avoided the problem of large clusters by limiting their size to 10. It is also possible to use decision rules that interpolate smoothly between retaining no large clusters and retaining all of them. This is the case of the rule originally implemented in LineUp whereby for each candidate pair of runs, the number of random pairs of runs with the same gene number but smaller SS value is tallied. This is divided by the number of pairs of runs in the bin or the number of permutations (generally 1000), whichever is bigger, to compute the significance of the pair of runs. With a threshold of $x/100$, this can be view as a simple decitions rule and it may not return a percentage $x/100$ of all the clusters, although in practice the difference is always very small due to the fact that large clusters are rare. Alternatively it can easily be modified as above to return exactly $x/100$ of all clusters.