BIOINFORMATICS

Vol. 15 no. 11 1999
Pages 909–917

Genome rearrangement with gene families

David Sankoff

Centre de recherches mathématiques, Université de Montréal, CP 6128 Succursale Centre-Ville, Montréal, Québec H3C 3J7, Canada

Abstract

Motivation: *The theory and practice of genome rearrangement analysis breaks down in the biologically widespread contexts where each gene may be present in a number of copies, not necessarily contiguous. In some of these contexts it is, however, appropriate to ask which members of each gene family in two genomes G and H, lengths l_G and l_H, are its true exemplars, i.e. which best reflect the original position of the ancestral gene in the common ancestor genome. This entails a search for the two exemplar strings of same length n (= number of gene families, including singletons), having the smallest possible rearrangement distance: the exemplar distance.*

Results: *A branch and bound algorithm calculates these distances efficiently when based on easily calculated traditional rearrangement distances, such as signed reversals distance or breakpoint distance, which also satisfy a property of monotonicity in the number of genes. Simulations show that in two random genomes, the expected exemplar distance/n is sensitive to the number and size of gene families, but approaches 1 as the number of singleton families increases. When the basic rearrangement distance is just the number of breakpoints, the expected cost of computing the exemplar breakpoints distance (EBD), as measured by total calls to the underlying breakpoint distance routine, is highly dependent on both n and the configuration of gene families. On the other hand, basing exemplar distance on exemplar reversals distance (ERD), the expected computing cost depends on the configuration of gene families but is not sensitive to n.*

Availability: *Code for EBD and ERD is available from the author or may be accessed at http://www.crm.umontreal.ca/~viart/exemplar_dis.html*

Contact: *sankoff@ere.umontreal.ca*

Introduction

The theory of genome rearrangements, exemplified by Hannenhalli and Pevzner's (1995a,b) polynomial algorithms for signed reversal distance and for translocation distance, and Caprara's (1997) NP-hardness result for unsigned reversal distance, takes for input two different orders of the same set of genes. These rearrangement distances measure the number of elementary operations necessary to transform one linear order on the genes into another, where the operations model genome-level evolutionary processes such as inversion (reversal) of a chromosomal segment, transposition of a segment from one site on a chromosome to another or translocation (exchange) of terminal segments between two chromosomes. [See Sankoff and Blanchette (1998) for a recent bibliography.]

Implicit in the notion that both genomes contain an identical set of genes is the assumption that homologies between all pairs of corresponding genes in the two genomes have previously been established. While this may be appropriate for some small genomes – most viruses and mitochondria, or for genomes of related species where long chromosomal segments, not just individual genes, correspond in the two genomes (cf. Nadeau and Sankoff, 1998) – it is clearly unwarranted for divergent species where several copies of the same gene, or several highly homologous (paralogous) genes may be scattered across the genome.

There have been a number of recent studies pertinent to the comparison of genomic sequences without the 'one copy of a gene per genome' assumption. Christie (1998) compares strings under reversals, where the restriction of one occurrence per genome of each alphabet symbol is relaxed. At present this analysis is constrained to a binary alphabet and to a formulation where the numbers of occurrences of a symbol is the same in both strings. Varré *et al.* (1999) compare genomic DNA sequences without any a priori reference to genes or other higher-level features of the genomes. This technique is based on the notion that one sequence is constructed by copying parts of the other, as often as necessary, and inserting unrelated sequence fragments elsewhere.

In this paper, we formulate a generalized version of the genome rearrangement problem where each gene may be present in a number of copies. The idea is to delete all but one member of each gene family – its *exemplar* – in each of the two genomes being compared, so as to minimize some rearrangement distance between the two reduced genomes thus derived.

To motivate this formulation, in Section 'Justification' we sketch a model in which genes may be repeatedly duplicated at any time during an ongoing process of genome rearrangement. For the case where the most recent com-

mon ancestor of the two genomes being compared contains exactly one member of each gene family, the optimal deletion problem stated above is directly justified as a way of inferring the total amount of rearrangement which differentiates the two given genomes. In Section 'The algorithm', we present a branch and bound algorithm for calculating *exemplar distances*. The keys to the feasibility of this approach are:

- an easily calculated distance, such as signed reversals distance or breakpoint distance,
- a property of monotonicity (as genes are inserted one-by-one) of these distances.
- a good strategy for the depth-first search.

We shall illustrate with two versions of the problem: exemplar breakpoints distance (EBD) and exemplar reversals distance (ERD). In Section 'Simulations' we use simulations to test the behavior of the solution as a function of the lengths of the input genomes and other problem parameters. We make use of the notion of *pegged genome*, previously defined in Section 'Pegged genomes', to ensure that all the simulations are comparable.

In Section 'Ancestral multigene families' we discuss the general case where the hypothetical ancestral genome is unconstrained as to how many members it may have in each gene family. We argue that the use of ancillary nucleotide sequence or amino acid sequence data provides us with a principled way of detecting such ancestral multigene families and of reducing the problem in this case to the original formulation of one ancestral gene per family. This is done through the expedient of considering the offshoots of each member in an ancestral family as constituting a separate family, for the purposes of the formal analysis. The characterization of gene families is essentially a statistical task of a phylogenetic nature and is not our immediate interest here; we show only how it is carried out in an idealized situation, i.e. purged of sampling variation.

We also explain, in Section 'The separability of the rearrangement and duplication analyses' why the internal analysis of gene families is formally separable from the calculation of the rearrangement distance, despite the fact that rearrangement and gene duplication actually occur in an interleaving manner over time. Once exemplars are identified, the locations of other members of a gene family shed no further light on the rearrangement process and, conversely, solution of the rearrangement problem does not help elucidate degrees of paralogy within a gene family.

To begin, we give some notation and definitions in Section 'The essential problem'.

The essential problem

Given an alphabet $\mathcal{A}$, let G and H be two strings (*genomes*) of signed (+ or −) symbols (representing *genes*) from $\mathcal{A}$, of lengths l_G and l_H, respectively. For each $a \in \mathcal{A}$, let $k_X(a)$ be the number of occurrences (+ or −) of symbol a in genome X. Without loss of generality, we may assume for all $a \in \mathcal{A}$, $k_G(a) > 0$ and $k_H(a) > 0$. All occurrences of the symbol a in both genomes are said to constitute a *gene family*, the 'a family'. For our purposes, that the genes in a family are not exact copies is immaterial; we simply assume that the families have been constructed correctly.

For each genome, an *exemplar* string is constructed by deleting all but one occurrence of each gene family. Call these g and h, respectively. Note that h is just a permutation of the symbols in g.

Consider two exemplar strings $g = g_1 \dots g_n$ and $h = h_1 \dots h_n$. Note that $n = |\mathcal{A}|$. We say g_i *precedes* g_{i+1} in g. If gene a precedes b in g and neither a precedes b nor $-b$ precedes $-a$ in h, they determine a *breakpoint* in g. Additional breakpoints are posited if $g_1 \neq h_1$ and if $g_n \neq h_n$. The *breakpoint distance* (BD) is the number of breakpoints in g, which is clearly equal to the number of breakpoints in h. The EBD between G and H is the minimum, over all choices of exemplar strings g and h, of the breakpoint distance between g and h.

A *reversal* transforms a string $\cdots x a \cdots b y \cdots$ to $\cdots x - b \cdots - a y \cdots$. The reversals distance (RD) between g and h is the minimum number of reversals necessary to transform g into h, or vice versa. The ERD between G and H is the minimum, over all choices of exemplar strings g and h, of the reversals distance between g and h.

EXAMPLE. Let $G = -b\ -a\ b\ a\ -c\ d\ c$, $H = a\ -a\ c\ a\ -c\ b\ d$. Based on the exemplar strings $-b\ -a\ -c\ d$ and $c\ a\ b\ d$, the EBD equals 2 and the ERD equals 1.

Justification

To explain the evolutionary pertinence of exemplar distances, we first give a simplified account of the divergence of a gene family. Suppose that in the most recent common ancestor F of genomes G and H, there is exactly one copy of gene a. After divergence over a time period of length t, the a family can grow (by gene duplication processes) from a single gene to several, independently in the two lineages leading to G and H. When a gene is duplicated, the copy may appear anywhere in the genome. (Processes resulting in tandem repeats may also occur.) In the meantime, each of the genomes is subject to rearrangement events, such as random reversals, occurring at any time up to time t.

One of the genes $a(G)$ in G is the *true exemplar*, or direct descendent of a, and has been moved from its original position in the genome by reversals (or other

rearrangement processes) only. The others are direct or indirect duplicates of $a(G)$ and were placed elsewhere in the genome during the duplication process. They may of course also have been moved around by the more recent reversals. Similarly, there is a true exemplar $a(H)$. The key idea for our method is that the true exemplars of each family will have been displaced marginally less frequently than the other members, and that this difference, cumulated over all the families, will result in the two true exemplar strings being distinctly less rearranged with respect to each other than any other pair of reduced genomes.

In the case of tandem repeats, if the two copies are subsequently separated by a rearrangement event, the one that is unaffected (i.e. which has moved less) continues to mark the current position of the true exemplar.

One measure of the evolutionary divergence of G and H is the number of breakpoints when comparing the subsequences of true exemplars in the two genomes. Another measure is the minimum number of reversals necessary to account for the difference in the order of the true exemplars in the two genomes. Of course, we do not know a priori which of the members of a family is the true exemplar. The most economical explanation is the one which requires the least rearrangement of the genome, hence the formulation of the two exemplar distance problems in Section 'The essential problem'.

Like any evolutionary inference method, one or more of the true exemplars may well be incorrectly identified using these criteria. And, as with any evolutionary reconstruction, such errors in themselves do not invalidate the method as a way of *estimating* the evolutionary divergence of the gene orders of the two genomes.

Pegged genomes

There are two distinct combinatorial approaches to calculating exemplar distance. One is to systematically examine the large number of possible exemplar strings resulting from all possible choices of pairs of exemplars, one in each genome, for all the gene families. The second is to generate all possible *different* exemplar strings in each genome before or concurrent with, the comparison of the two genomes. The latter approach would clearly be preferable if each possible exemplar string were *ambiguous*, i.e. could be produced by very many different choices of exemplar pairs. Some genomes do contain regions where large numbers of copies from a few gene families are closely intermingled, but this is the exception; most multi-gene families tend to be scattered throughout the genome and separated by long stretches of single-copy genes, so that each exemplar string can be constructed in only one way. Thus the algorithm we propose in Section 'The algorithm' attempts no economies based on avoiding repeated trials of the same ambiguous exemplar string.

Though the algorithm is nonetheless fully applicable to the case where there are ambiguities, in the simulations in Section 'Simulations', we will use only unambiguous data, to assure comparability of the various experiments, as explained below. To construct these genomes, we first characterize them as follows.

A gene is a *singleton* in a genome if it is the only member of its family in that genome. A *single-pair family* consists of two singletons, one in each genome. The remaining families are *multi-pair*, even if they are only represented by a singleton in one of the genomes. A genome is said to be *pegged* if between any two members of a gene family, there is at least one singleton.

THEOREM 1. *The number of different pairs of exemplar strings is*

$$N \leq \prod_{\text{gene families } a} k_G(a) k_H(a)$$

where equality holds if and only if both genomes are pegged.

PROOF. The formula for N counts the number of ways of constructing an exemplar string by choosing all possible exemplar pairs from each family.

To show that equality holds for pegged genomes, it suffices to show that any exemplar string extracted from a pegged genome is consistent with only one member from each family. Suppose the contrary, that some exemplar string could be constructed using either member $a(1)$ or $a(2)$ of family a. Between these two, there must be some singleton gene x. Thus any exemplar string constructed with $a(1)$ must be different from any constructed with $a(2)$, a contradiction.

Now suppose that at least one of the genomes is not pegged. Then for some gene family a, there is an $a(1)$ and an $a(2)$ separated by as small a number of genes as possible, none of them a singleton. Because of this minimality condition, no two genes from the same family can be between $a(1)$ and $a(2)$. Then either $a(1)$ and $a(2)$ are adjacent or are separated by at most one gene from each of one or more families. Consider an exemplar string containing none of these intervening genes (if there are any). Such a string is always possible because none of the intervening gene are singletons. But the same string can be constructed using either $a(1)$ or $a(2)$. So there are fewer than N different pairs of exemplar strings. □

The algorithm

Lower bound

Suppose we delete all occurrences of gene family a from genomes G and H. If an exemplar distance must then

decrease or remain the same, we say it is *monotonic*. The following is readily verified:

THEOREM 2.

(a) EBD is monotonic.

(b) ERD is monotonic.

Thus both exemplar distances for G and H are bounded below by the corresponding distances for the genomes resulting when all symbols from some subset of $\mathcal{A}$ are deleted. The monotonicity property allows us to construct the exemplar strings one gene at a time, efficiently pruning unpromising parts of the search tree.

The distances

The breakpoint distance, introduced by Watterson *et al.* (1982), is easily calculated in time linear in the length of the genomes. The signed reversals distance (Sankoff, 1989; Kececioglu and Sankoff, 1994), first proved to be of polynomial complexity by Hannenhalli and Pevzner (1995a), can also be calculated in near-linear time (Berman and Hannenhalli, 1996; Kaplan *et al.*, 1997), though this requires coding of a very intricate algorithm. Note that we require only the calculation of the reversals distance and not the recovery of the actual reversals, which would take much longer.

Strategy

Our algorithm for calculating the exemplar distances focuses on one gene family at a time, choosing the pair of exemplars which least increases the distance when inserted into to the partial exemplar string already constructed. The monotonicity property ensures that no pair can decrease this distance, so that whenever the distance between partial exemplar strings attains that of the current best complete exemplar strings, these partially constructed strings, as well as more complete strings based on them, need not be examined further.

In what order should the gene families be searched? Our strategy is to start with the families of size two, i.e. those with only one exemplar in each genome, then those of size three, four and so on. This is the fastest way to build long partial exemplar strings so that pruning by means of the lower bound can come into play as soon as possible, and the combinatorial explosion due to large families may be avoided.

Branch and bound

The algorithm presented below starts with an initial *partial exemplar string* for each genome consisting of the genes from the single-pair families *in the same linear order as they are in the original genomes*. Each partial exemplar string is then expanded with a gene from the next-smallest (as measured by $k_G(\cdot)k_H(\cdot)$) family, as follows. All possible pairs in the family are tested to see how much they increase the exemplar distance when the two members are inserted into the partial exemplar strings, *at the points determined by the linear orders in the original genomes*. The test values are stored (sorted according to size). All the pairs are initially in an *unused* state. The partial exemplar string is expanded through the inclusion of the pair minimizing the increase in exemplar distance, which is declared *used*. The next-smallest family is then examined, and so on.

Note that the exemplar string is not constructed from left-to-right or from right-to-left, but rather by inserting the two genes of a pair into the partial exemplar strings at the (unique) positions consistent with the linear orders in the original genomes.

A backtracking step from the family currently being considered occurs whenever it contains no remaining unused pairs whose test values are small enough, i.e. all the unused pairs would increase the exemplar distance beyond the current best value. The exemplar pair from this family is implicitly deleted from the partial exemplar string (but remains in the 'used' state). When backtracking leads to a family containing unused pairs with low enough scores, backtracking ceases, and the expansion of the partial exemplar string resumes with the addition of the pair minimizing the test score, which assumes used status.

Note that all the gene pairs in a family are tested, and declared unused, only when this family is reached through a forward, or potential expansion step. Retesting is unnecessary when the family is reached through a backtracking step, because the stored values are still valid. This is an important device for saving computation time, since it is the test which calls the BD or RD distance routine.

An experimental implementation of the algorithm makes use of Hannenhalli's (1995) code for calls to the RD calculation.

Algorithm find_exemplars

input genomes G and H.

Label gene families from 1 to $n = |\mathcal{A}|$ in order of increasing $k_G(\cdot)k_H(\cdot)$. Form partial exemplar strings from genes in the s single-pair families, ordered as in G and H.

In all families, all pairs of genes are initially *unused*.

$i \leftarrow s + 1$.

current_distance $\leftarrow \infty$.

while $i \geq s$

if there remain unused pairs in the i-th family

if all the pairs in the i-th family are unused

For each pair, evaluate the distance d between the partial exemplar strings constructed by inserting this pair into the two partial exemplar strings based on the first $i-1$ families.

Store the distances associated with all pairs, sorted by size.

end if

Expand the partial exemplar strings by inserting the unused pair with minimal d. This pair acquires 'used'status.

if $d \geq$ current_distance

All pairs in the i-th family regain unused status.

$i' \leftarrow i-1$.

else

$i' \leftarrow i+1$.

end if

else

All pairs in the i-th family regain unused status.

$i' \leftarrow i-1$.

end if

if $i = n$

current_distance $= d$.

$i' \leftarrow i-1$.

end if

$i \leftarrow i'$.

end while

output exemplar strings, d

stop

Simulations

Though our algorithm functions for either pegged or unpegged genomes, it can be very inefficient for the latter, especially when there are many highly ambiguous exemplar strings. Thus in our simulations, we will use only pegged genomes, so that each exemplar string is derived in a unique way, a condition that better approximates biological realism. This will enable us to precisely assess the computational savings obtained by the branch and bound algorithm, since we know that a straightforward exhaustive evaluation of all pairs of potential exemplar strings would require exactly N calls to the rearrangement distance routine. Indeed, we will measure the computational cost of our solutions in terms of the number of calls to the distance routine. The performance of these routines when optimally programmed is already known (linear for BD, somewhat more than linear for RD), so we will not evaluate it here.

Though N is the appropriate benchmark for computational cost, there are other parameters of the input genomes which have a greater effect on both this cost and the expected exemplar distance between random genomes. These parameters include the number of single-pair gene families and the number and size of multi-pair families.

For comparability, then, all our simulations will involve the same (large) value of N. Each experiment will be based on a fixed configuration of multi-pair families consistent with this value of N, and 100 pairs of random pegged genomes will be evaluated for each of a series of exemplar lengths n. The genomes will be generated at random, as determined by a uniform probability over all genomes with the given configuration of multi-pair families and given exemplar length, conditioned by the pegging requirement, as described in Section 'The pegged genome generator'.

Each pair will be evaluated by the algorithm *find_exemplars*, and both the final exemplar distance and the total number of calls to the distance (BD or RD) routine, as a measure of computational cost, are to be noted.

The pegged genome generator

For the genomes of a given exemplar length containing a given configuration of multi-pair families to be chosen according to a uniform probability rule, conditioned by the pegging requirement, the following procedure suffices.

- Construct an initial partial genome by generating a random permutation of the σ singleton genes.
- The genes from the multi-gene families are inserted into the partial genome one family at a time, as follows. The genes in the current family are inserted one at a time. A gene is inserted with equal probability at any position, namely between any two adjacent genes already in the partial genome, or preceding the first gene or following the last one, except for *blocked* positions. When a gene is inserted, all positions in the interval between the closest singletons to its left and its right are blocked for the remaining genes in the current family. (Or all positions preceding the first singleton or following the last singleton, if that is where the gene is inserted.)

Configurations of multi-pair families

Recall that the lengths of genomes G and H are l_G and l_H, and the lengths of the exemplar strings is $n = |\mathcal{A}|$. Let s and m be the number of single-pair and multi-pair families, respectively, where $s < \min\{\sigma_G, \sigma_H\}$, the number of singletons in the two genomes. Finally,

Table 1. Number and sizes of multi-pair gene families used in simulations, all with $N \approx 4 \times 10^9$. Vertically aligned numbers in G and H refer to the same family. Simulations varied according to number of single-pair families, which does not affect N

Label	Genome	Size of families	N
Unbalanced	G	10 5 5 2 2 2 2 2 2 2 2 1 1 1 1	4.096×10^9
	H	10 5 5 2 2 2 2 1 1 1 1 2 2 2 2	
Nines	G	10 9 9 9 9	4.305×10^9
	H	9 10 9 9 9	
Sixes	G	7 7 6 6 6 6	4.033×10^9
	H	6 6 7 7 6 6	
Threes	G	5 2 3 3 3 3 3 3 3 3 3	4.305×10^9
	H	2 5 3 3 3 3 3 3 3 3 3	

let $v_X = \sum_{\text{gene families } a} k_X(a)$ the number of genes in genome X that are in multi-pair families, for $X \in \{G, H\}$.

We have

$$l_X = v_X + s$$

and

$$n = m + s,$$

so that

$$l_X = n + v_X - m.$$

For a given configuration of multi-pair families, any change in the length of the exemplar string is reflected in identical changes in the length of each genome.

To simplify the design of our experiments, we will confine ourselves to configurations where $v_G = v_H$, so that $l_G = l_H$ as well. In Table 1, we present the configurations used in the simulations.

Distances

How does the configuration of multi-pair families affect the average exemplar distance (as a function of exemplar string length) and the computation time necessary to find it? Figure 1 shows how, for each of the configurations of gene families, the mean EBD and ERD increase as genome length increases. For comparability, the exemplar distances are normalized by the exemplar length n, since the expected value of the breakpoint distance for random permutations of length n is $n - \frac{1}{2}$ and that of the reversals distance is also very close to n (Kececioglu and Sankoff, 1994). Note that to peg a genome X, the minimum number of singletons required is given by

$$\sigma_X \geq \max_{\text{gene families } a} k_X(a) - 1,$$

limiting the leftward extent of the curve for each configuration of multi-pair families. The rightward extent was limited by computing time considerations.

It can be seen that for small genomes, the exemplar distance, even when normalized, is much smaller than that of random genomes of the same length n, but this effect tends to disappear as G and H contain more and more single-pair families, i.e. as s and l increase.

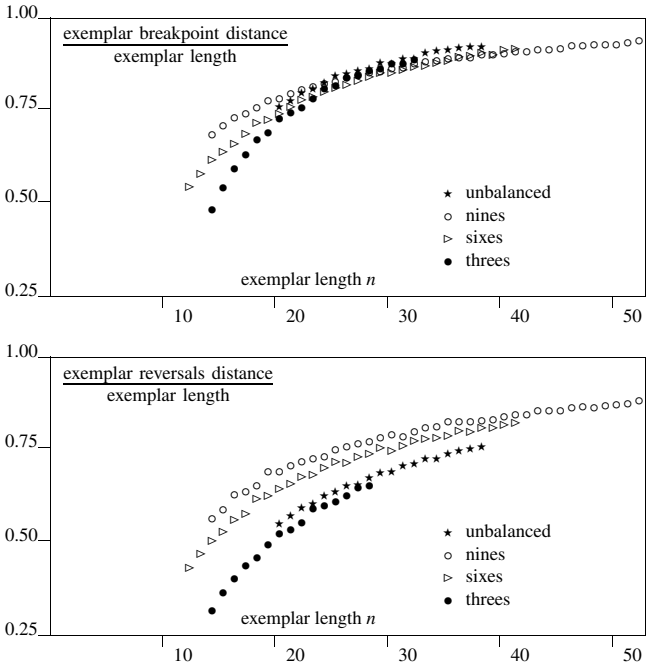

Fig. 1. Normalized average exemplar distances (sample size 100) as a function of exemplar string length for configurations of multi-pair families in Table 1.

Another observation is that for a given exemplar string of length n the exemplar distance tends to be larger, at least for smaller n, for configurations consisting of few large families than for those made up of many small families. This effect also disappears as s and l increase, especially for EBD. The 'unbalanced' configuration, made up largely of families with only two or four pairs, behaves like a small-family configuration.

Costs

Turning to the computational cost of the algorithm, Figure 2 shows dramatic difference between EBD and ERD and among the different configurations. Considering the logarithmic scale, the differences are even greater. The configurations with few, large, families, are rapidly analyzed for the EBD, while those with many small families take much longer, increasingly so for larger n. With the large families, it is often the case that the true exemplars are found in the initial greedy accumulation of the best pairs in each family, and that these exemplars do not add any incremental distance to the value calculated with single-pair families only. In this case, the lower bound procedure succeeds in exempting any further calls

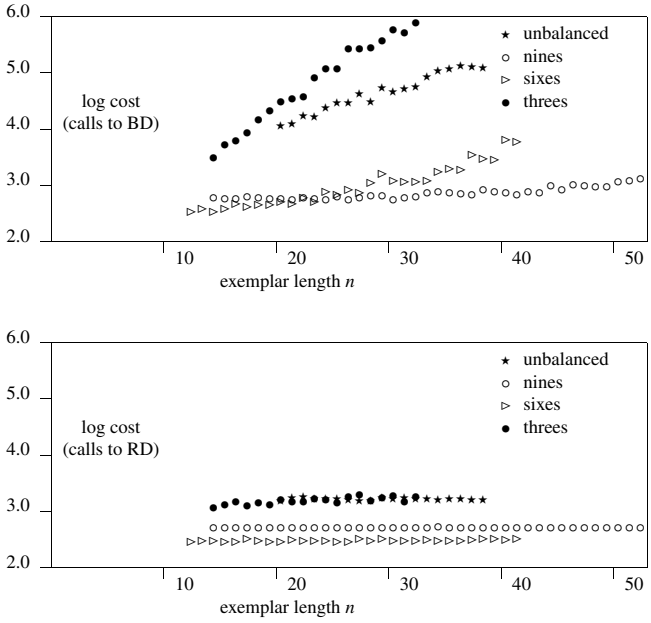

Fig. 2. Average log cost (sample size 100) in terms of number of calls to breakpoint (top) or reversals (bottom) distance calculation as a function of exemplar string length for configurations of multi-pair families in Table 1.

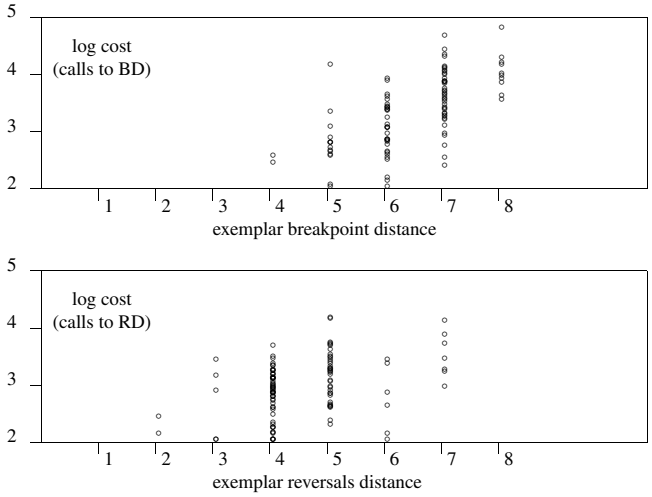

Fig. 3. Log of the number of calls to distance routines as a function of exemplar distance for 100 pairs of sample genomes with the configuration of 'threes' and four single-pair families ($n = 14$ and $l = 35$).

to the distance routine.

Figure 2 also shows that in comparison to the EBD which becomes very costly for large s and l, remarkably the branch and bound algorithm seems as efficient in calculating ERD for longer genomes as it is for short ones (recall that N is fixed), though different multi-pair family configurations require different amounts of computing.

We note as well that in comparison to our exhaustive search benchmark of $N \approx 4 \times 10^9$, the branch and bound procedure is extremely efficient. The mean log cost was less than 6 in all our simulations, though calculating EBD with the 'threes' and 'unbalanced' configurations occasionally required in excess of 10^7 calls to the BD routine, and once, more than 10^8.

Cost versus distance

Within a sample of 100 simulated genomes having the same configuration of multi-pair families and the same number of single-pair families, what is the distribution of exemplar distances, the distribution of computational costs and the connection between these two variables? Figure 3 depicts the answer to these questions in a typical case.

We note that the cost is extremely variable, but that there is a clear tendency for solutions to be more rapidly found for smaller exemplar distances.

Ancestral multigene families

Suppose that in the most recent common ancestor F of genomes G and H, there are already $k = k_F(a) > 1$ copies of gene a. Label these $a_1, \ldots, a_k$. Were we to directly seek a true exemplar string based on G and H, only one member of the a family would be chosen in each genome, whereas there should be k of them, to reflect the original configuration in the ancestor F. For analytical purposes, then, we would like to consider the descendants of each a_i, including all those in G and those in H, to constitute a distinct gene family, though from a biological point of view they are of course related.

Suppose we could measure the divergence time between two homologous (paralogous or orthologous) genes exactly, e.g. by DNA sequence comparison. Then if we examine the set of divergence times among the members of all the a_i gene families, $i = 1, \ldots, k$, in both G and H, we could distinguish three groups of results.

1. Those pairs of genes with divergence times greater than t derive from different genes a_i and a_j in F, where $i \neq j$. (There are none if $k_F(a) = 1$.)

2. Those pairs of genes with divergence times less than t must be within the same genome, and must have diverged since the split of the lineages of G and H, i.e. they are in the same a_i family.

3. The remaining pairs of genes, with divergence times exactly t, must consist of one from G and one from H, and derive from a single gene a_i in F.

We may thus consider the a_i families distinct for the purposes of the EBD and ERD problems, despite their ancient homology. What is important is that we can distinguish between them by their early divergence times, and so we can investigate the changing position of their exemplars without ambiguity, as we do with exemplars of unrelated families. Without loss of generality, then, we may assume that $k_F(a) = 1$ for all gene families a, so that $l_F = n$.

The separability of the rearrangement and duplication analyses

Of course, we cannot measure divergence times exactly. This may become a source of error in the input to the exemplar distance calculation. Nevertheless, it is the idea of seeking the rearrangement history of the true exemplar descendants of the ancestral genes which induces a decomposition of the rearrangement problem with gene families into two stages, the first to identify the families, including possibly some families (or subfamilies) which are themselves remotely homologous, and second the exemplar distance calculation.

Because we have put no constraints on where duplicates genes appear in the genome, the evolutionary relationship among the genes in a family does not contain information about the rearrangement history of the true exemplar strings. And conversely this history does not constrain the paralogous relationships in any way.

The phylogenetic methodology dates duplication events in absolute time according to an appropriate clock. The inferred rearrangement events can be assigned times along this scale arbitrarily, as long as they are ordered in the right sequence. For example they may be placed at equal intervals from time 0 to time t. For genome G, say, a duplicate gene originating at time $0 < u < t$ can be re-inserted into the intermediate genome between F and G immediately before time u, appropriately adjacent to (and with the appropriate sign) the same gene it is eventually adjacent to in G. The remaining rearrangement events can always be slightly redefined to assure that this adjacency is never interrupted. Whatever the inferred duplication and rearrangement histories, the two are compatible.

It is possible to define an interesting and realistic variant of the problem where the two components would have to be solved in concert, namely one where not only genes but entire fragments of the genome could be duplicated at a single step. New methods would be required in this case.

Discussion

To justify our method, we have assumed a relatively unconstrained model of gene family proliferation, together with a restrictive, biological clock model of sequence divergence. But perhaps the most serious deficiency of this model is its neglect of gene loss. This phenomenon, whether due to actual physical deletion or rapid divergence to pseudogene status, threatens the validity of the method insofar as it might be the true exemplars which are lost. It would not be difficult to incorporate gene loss into a more general formulation of the exemplar distance problem, but this would necessarily involve weights or other arbitrary devices to prevent trivial solutions. There is, however, an independent approach to the question of gene duplication and gene loss which might be combined with the present analysis to remove some of the arbitrariness of both. This is the *reconciliation* model which is used to account for anomalies in gene trees when they are compared to species trees (Page and Charleston, 1997). Constraining analyses to take into account both gene order and reconciliation may represent a productive approach.

There are a number of specialized theories of gene family origins, maintenance and evolution. To the extent that these hold true, they may complicate or invalidate the present approach. For example, gene families may arise through various gene duplication schemes (e.g. Altenberg, 1995), genome duplication (Nadeau and Sankoff, 1997; Wolfe and Shields, 1997) or hybridization (El-Mabrouk and Sankoff, 1999). It would seem feasible to allow for these in the present approach. On the other hand, there is speculation that small gene families ($k \leq 6$) are mostly primordial, i.e. have been maintained from the earliest times (Slonimski *et al.*, 1998), so that the notion of exemplars would lose its significance.

Whether or not the reconciliation analysis is pertinent, the phylogenetic context represents an important direction for the generalization of our theory, in the first instance through the investigation of a *median exemplars problem* in analogy to the median breakpoint problem (Sankoff and Blanchette, 1997), the median rearrangements problem (Sankoff *et al.*, 1996; Caprara, 1999) and original synteny (Ferretti *et al.*, 1996). Another avenue of generalization would involve exemplar distance problems formulated for translocation distance (Hannenhalli and Pavzner, 1995b) or syntenic distance (Ferretti *et al.*, 1996; Liben-Nowell, 1999). From the combinatorial viewpoint, the entire question of efficient calculation of exemplar distances for unpegged genomes remains open.

Our simulations indicate that the applicability of our methodology is highly dependent on the context. Where there are just a few gene families, even if these have many members, the size of genome that can be handled approaches that of the underlying traditional rearrangement distance calculation. When there are many gene families, even if they are small, the branch and bound approach to the EBD calculation would seem to break down for large genomes. It is in this case, however, that alternative algorithmic approaches may also be feasible. The branch and bound algorithm for the ERD measure, on the other

hand, is little affected by genome size as long as N is constant. Of course the RD calculation itself depends more than linearly on genome length.

Acknowledgements

Research supported by grants from the Natural Sciences and Engineering Research Council of Canada and the Canadian Genome Analysis and Technology program. Hannenhalli's program was adapted by Mathieu Blanchette and Sylvain Viart, and the latter also coded the pegged genome generator and carried out the simulations. Thanks to Nadia El-Mabrouk for discussions of the RD algorithm and of the general (unpegged) exemplar distance problem. The author is a Fellow in the Evolutionary Biology Program of the Canadian Institute for Advanced Research.

References

Altenberg,L. (1995) Genome growth and the evolution of the genotype–phenotype map. In Banzhaf,W. and Eeckman,F.H. (eds). *Evolution and Biocomputation: Computational Models of Evolution. Lecture Notes in Computer Science* **899**. Springer, New York, pp. 205–259.

Berman,P. and Hannenhalli,S. (1996) Fast sorting by reversal. In Hirschberg,D. and Myers,G. (eds). *Combinatorial Pattern Matching. 7th Annual Symposium. Lecture Notes in Computer Science* **1075**. Springer, New York, pp. 168–185.

Caprara,A. (1997) Sorting by reversals is difficult. In *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB 97)* ACM, New York, pp. 75–83.

Caprara,A. (1999) Formulations and hardness of multiple sorting by reversals. In Istrail,S., Pevzner,P. and Waterman,M. (eds). *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB 99)* ACM, New York, pp. 84–93.

Christie,D.A. (1998) Sorting strings by global transformations. Chapter 5 of *Genome Rearrangement Problems*. PhD dissertation, Department of Computing Science, University of Glasgow.

DasGupta,B., Jiang,T., Kannan,S., Li,M. and Sweedyk,Z. (1997) On the complexity and approximation of syntenic distance. In *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB 97)* ACM, New York, pp. 99–108.

El-Mabrouk,N. and Sankoff,D. (1999) Hybridization and genome rearrangement. In Crochemore,M. and Paterson,M. (eds). *Combinatorial Pattern Matching. 10th Annual Symposium. Lecture Notes in Computer Science* **1645**. Springer, New York, pp. 78–87.

Ferretti,V., Nadeau,J.H. and Sankoff,D. (1996) Original synteny. In Hirschberg,D. and Myers,G. (eds). *Combinatorial Pattern Matching. 7th Annual Symposium. Lecture Notes in Computer Science* **1075**. Springer, New York, pp. 159–167.

Hannenhalli,S. (1995) Program for computing an optimal sorting by reversal for signed permutations.http://www-hto.usc.edu/software/distance/signed_dist.c.

Hannenhalli,S. and Pevzner,P.A. (1995a) Transforming cabbage into turnip. (polynomial algorithm for sorting signed permutations by reversals). In *Proceedings of the 27th Annual ACM-SIAM Symposium on the Theory of Computing* pp. 178–189.

Hannenhalli,S. and Pevzner,P.A. (1995b) Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of the IEEE 36th Annual Symposium on Foundations of Computer Science* pp. 581–592.

Kaplan,H., Shamir,R. and Tarjan,R.E. (1997) Faster and simpler algorithm for sorting signed permutations by reversals. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms* ACM, New York, pp. 344–351.

Kececioglu,J. and Sankoff,D. (1994) Efficient bounds for oriented chromosome inversion distance. In Crochemore,M. and Gusfield,D. (eds). *Combinatorial Pattern Matching. 5th Annual Symposium. Lecture Notes in Computer Science* **807**. Springer, New York, pp. 307–325.

Liben-Nowell,D. (1999) On the structure of syntenic distance. In Crochemore,M. and Paterson,M. (eds). *Combinatorial Pattern Matching. 10th Annual Symposium. Lecture Notes in Computer Science* **1645**. Springer, New York, pp. 50–65.

Nadeau,J.H. and Sankoff,D. (1997) Comparable rates of gene loss and functional divergence after genome duplications early in vertebrate evolution. *Genetics*, **147**, 1259–1266.

Nadeau,J.H. and Sankoff,D. (1998) Counting on comparative maps. *Trends Genet.*, **14**, 495–501.

Page,R. and Charleston,M.A. (1997) From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem. *Mol. Phylogenet. Evol.*, **7**, 231–240.

Sankoff,D. (1989) Mechanisms of genome evolution: models and inference. *Bull. Int. Stat. Instit.*, **47**, 461–475.

Sankoff,D. and Blanchette,M. (1997) The median problem for breakpoints in comparative genomics. In Jiang,T. and Lee,D.T. (eds). *Computing and Combinatorics, Proceedings of COCOON '97. Lecture Notes in Computer Science* **1276**. Springer, New York, pp. 251–263.

Sankoff,D. and Blanchette,M. (1998) Multiple genome rearrangement and breakpoint phylogeny. *J. Computat. Biol.*, **5**, 555–570.

Sankoff,D., Sundaram,G. and Kececioglu,J. (1996) Steiner points in the space of genome rearrangements. *Int. J. Foundations Comput. Sci.*, **7**, 1–9.

Slonimski,P.P., Mossé,M.-O., Golik,P., Hénault,A., Diaz,Y., Risler,J-.L., Comet,J.-P., Aude,J.-C., Wozniak,A., Glémet,E. and Codani,J.-J. (1998) The first laws of genomics. *Microb. Comp. Genomics*, **3**, 46.

Varré,J.-S., Delahaye,J.-P. and Rivals,É. (1999) Transformation distances: a family of dissimilarity measures based on movements of segments. *Bioinformatics*, **15**, 194–202.

Watterson,G.A., Ewens,W.J., Hall,T.E. and Morgan,A. (1982) The chromosome inversion problem. *J. Theor. Biol.*, **99**, 1–7.

Wolfe,K.H. and Shields,D.C. (1997) Molecular evidence for an ancient duplication of the entire yeast genome. *Nature*, **387**, 708–713.