6

Estimating Similarities in DNA Strings Using the Efficacious Rank Distance Approach

Liviu P. Dinu[1] and Andrea Sgarro[2]
[1]*University of Bucharest, Faculty of Mathematics and Computer Science, Academiei, Bucharest*
[2]*University of Trieste + CBM, Area Science Park, Trieste*
[1]*Romania*
[2]*Italy*

1. Introduction

In general, when a new DNA sequence is given, the first step taken by a biologist would be to compare the new sequence with sequences that are already well studied and annotated. Sequences that are similar would probably have the same function, or, if two sequences from different organisms are similar, there may be a common ancestor sequence.

Traditionally, this is made by using a *distance function* between the DNA chains, which implies in most cases that we apply it between two DNA sequences and try to interpret the obtained score. The standard method for sequence comparison is by sequence alignment. Sequence alignment is the procedure of comparing two sequences (pairwise alignment) or more sequences (multiple alignment) by searching for a series of individual characters or characters patterns that are in the same order in the sequences. Algorithmically, the standard pairwise alignment method is based on dynamic programming; the method compares every pair of characters of the two sequences and generates an alignment and a score, which is dependent on the scoring scheme used, i.e. a scoring matrix for the different base-pair combinations, match and mismatch scores, or a scheme for insertion or deletion (gap) penalties. The underlying string distance is called *edit distance* or also *Levenshtein distance*.

Although dynamic programming for sequence alignment is mathematically optimal, it is far too slow for comparing a large number of bases. Typical DNA database today contains billions of bases, and the number is still increasing rapidly. To enable sequence search and comparison to be performed in a reasonable time, fast heuristic local alignment algorithms have been developed, e.g. BLAST, freely available at http://www.ncbi.nlm.nih.gov/BLAST.

With respect to the standard approach to the alignment and string matching problems as dealt with in computer science, alternative approaches might be explored in biology, provided one is able to give a positive answer to the following question: can one exhibit a sequence distance which is at the same time easily computed and non-trivial? The ranking of this problem on the first position in two lists of major open problems in bioinformatics (J.C. Wooley. Trends in computational biology: a summary based on a RECOMB plenary lecture. *J. Comput. Biology*, 6, 459-474, 1999 and E.V. Koonin. The emerging paradigm and open problems in comparative

genomics. *Bioinformatics*, 15, 265-266, 1999) are reasons enough to say that the DNA sequence comparison is actually an exciting problem which has been long waiting for new approaches.
We present in this chapter new alternatives: first, we introduce a low-complexity but non-trivial distance for strings, called rank distance, and advocate its use in biology; we give preliminary experimental validations on biological data. Our method is easy to implement, does not use the standard alignment principle, and has an extremely good computational behavior. Another advantage of our method is that it imposes minimal hardware demands: it runs in optimal conditions on modest computers, reducing the costs and increasing the number of possible users. So, for example, the time needed to compare a DNA string of 45.000 nucleotides length with other 150 DNA strings (with similar length), by using an laptop with 224 MB RAM and 1.4 GHz processor is no more than six seconds.
To measure the distance between two strings, we use the following strategy: we scan (from left to right) both strings and for each letter from the first string we count the number of elements between its position in first string and the position of its first occurrence in the second string. Finally, we sum all these scores and obtain the rank distance.
Clearly, the rank distance gives a score zero only to letters which are in the same position in both strings, as Hamming distance does (we recall that Hamming distance is the number of positions where two strings of the same length differ). On the other hand, an important aspect is that the reduced sensitivity of the rank distance w.r. to deletions and insertions is of paramount importance, since it allows us to make use of *ad hoc extensions to arbitrary strings*, such as do not affect its low computational complexity, unlike what happens with the extensions of the Hamming distance, mathematically optimal but computationally too heavy, which lead to the *edit-distance*, or *Levenshtein distance*, and which are at the base of the standard alignment principle. So, the rank distance sides with Hamming distance rather than Levenshtein distance as far as computational complexity is concerned: the fact that in the Hamming and in the rank case the median string problem is tractable, while in the edit case it is is NP-hard, is a very significant indicator.
Rank distance, as most tools of this kind, provides an evaluation which is strictly limited to the closeness of two objects in themselves, without looking at the relation of the respective objects with the rest of universe. We take in this work also an alternative point of view, such as not to neglect the relation which the two objects (the two DNA strings) bear with respect to the universe of all string we are considering (even if the absolut distance is large, it may relatively small, when the two objects are situated in a sparse area of the universe, with many "holes" and with few "neighbour" strings. We make use of the notion of distinguishability between two sequences (basically a Shannon-theoretic notion closely related to the problem of the median) and we investigate it in the metric space of rank distances, thinking of application in biology, first of all in DNA word design, a form of coding for biological computers where the codewords are DNA strings. In a way, the rank distance is local, as most distances are, while the rank distinguishability is a global non-local variant thereof. Actually, we think that this is in line also with more traditional bioinformatics problems: when a new DNA sequence is given, the first step taken by a biologist is to compare the new sequence with sequences that are already well studied and annotated. Sequences that are similar would probably have the same function, and the similarity of two sequences should be related to the sparseness of the neighbouring space of strings.
The tractability of the median string via rank distances offers us the possibility to investigate a well studied theme: to investigate the median and the centre of a given multiset of sequences by using our "global" approach. At this stage, there appear to be still many open problems, and the results we are able to exhibit are just preliminary. Also, we can use this approach to

deal with a multi-combining categorization schema, in the sense of multi-agents. The research is supported by new experimantal data.

2. Rank distance

To measure the distance between two strings, we use the following strategy: we scan (from left to right) both strings and for each letter from the first string we count the number of elements between its position in first string and the position of its first occurrence in the second string. Finally, we sum all these scores and obtain the rank distance.

Initially, rank distances between strings were used in computational linguistics, cf. (13), (14); later their use was extended to such application domains as is bioinformatics, cf. (17), or authorship identification, cf. (20). The reasons for the practical success of the rank distance are basically two:

i) it is quite *quick* to compute

ii) it is *robust* with respect to small modifications of the sequences

As for the first point, the computational effort is only linear in the sequence length n, as happens with the unsophisticated *Hamming distance*, but unlike what happens with the more sophisticated *edit distance*, whose computational complexity is quadratic in n, cf. (8). As for the second point, think e.g. of a sequence x and “by mistake" rotate it one position to the right to get y: if, say, x is 01 repeated $n/2$ times the Hamming distance between x and y is as high as n, and so the percentage “error" scaled to the maximal possible value for Hamming distances is as high as 100%. Instead, while the rank distance is still linear in n, the percentage error goes to zero with the sequence length n, and so is practically negligible for very long sequences as are, say, DNA strings.

In other words, the rank distance measures the "gap" between the positions of a letter in the two given strings, and then add these measures. In a way, we can say that the rank distance gives us the total non-alignment score between two sequences.

Clearly, the rank distance gives a score zero only to letters which are in the same position in both strings, as Hamming distance does (we recall that Hamming distance is the number of positions where two strings of the same length differ). On the other hand, an important aspect is that the reduced sensitivity of the rank distance w.r. to deletions and insertions (cf. the paragraph at the end of this subsection) is of paramount importance, since it allows us to make use of *ad hoc extensions to arbitrary strings*, such as do not affect its low computational complexity, unlike what happens with the extensions of the Hamming distance, mathematically optimal but computationally heavy, which lead to the *edit-distance*, or *Levenshtein distance*, and which are at the base of the standard alignment principle. So, rank distance sides with Hamming distance rather than Levenshtein distance as far as computational complexity is concerned: the fact that in the Hamming and in the rank case the median string problem is tractable (16), while in the edit case it is is NP-hard (23), is a very significant indicator.

2.1 Preliminaries and formal definitions

The rank distance is an *ordinal* distance tightly related to the so-called *Spearman's footrule* [1], which has long been used in non-parametric statistics. However, the latter evaluates derangements between permutations of the integers $1, 2, \ldots, n$, rather than distances

[1] Both Spearman's footrules and binary Hamming distances are a special case of a well-known metric distance called sometimes taxi distance, which is known to be equivalent to the usual Euclidian distance. Computationally, taxi distance is obviously linear.

(dissimilarities) between sequences on a *fixed and known alphabet*, as is the case with rank distances [2]. Unlike other ordinal distances, the Spearman's footrule is linear in n, and so very easy to compute. Its average value is at two-thirds of the way to the maximum value (both are quadratice in n); this is because, in a way, the Spearman footrule becomes rather "undiscriminating" for highly different orderings. Rank distance has the same drawbacks and the same advantages of Spearman's foootrule. As for "classical" ordinal distances for integers, with averages values, maximal values, etc., the reader is referred to the basic work (11).

Let us go back to strings. Let us choose a finite alphabet, say $\{A, C, G, T\}$ as relevant for DNA strings, and two strings on that alphabet, which for the moment will be constrained to be a permutation of each other (i.e. they have the same composition or the same Parikh vector). E.g. take the two strings of length 6, $AACGTT$ and $CTGATA$; number the occurrences of repeated letters in increasing order to obtain $A_1A_2C_1G_1T_1T_2$ and $C_1T_1G_1A_1T_2A_2$. Now, proceed as follows: in the first sequence A_1 is in position 1, while it is in position 4 in the second sequence, and so the difference is 3; compute the difference in positions for all letters and sum them. In this case the differences are 3, 4, 2, 1, 3, 1 and so the distance is 14. Even if the computation of the rank distance as based directly on its definition may appear to be quadratic, we shall exhibit below two algorithms which take it back to linear complexity.

In computational linguistics the rank distance for strings *without repetitions* had been enough. In a way, *indexing* converts a sequence *with repetitions* into a sequence without repetitions, in which the k occurrence of a letter a are replaced by single occurrences of the k indexed letters a_1, a_2, ..., a_k. Let $u = x_1x_2 \dots x_n$ and $v = y_1y_2 \dots y_m$ be two strings of lengths n and m, respectively. For an element $x_i \in u$ we define its *order* or *rank* by $ord(x_i|u) = i$: we stress that the rank of x_i is its position in the string, counted from the left to the right, *after* indexing, so that for example the second T in the string $CTGATA$ has rank 5.

Note that some for arbitrary strings (indexed) occurrences appear in both strings, while some other are *unmatched*, i.e. they appear only in one of the two strings. In definition (1) the last two summations refer to these unmatched occurrences. More precisely, the first summation on $x \in u \cap v$ refers to occurrences x which are common to both strings u and v, the second summation on $x \in u \setminus v$ refers to occurrences x which appear in u but not in v, while the third summation on $x \in v \setminus u$ refers to occurrences x which appear in v but not in u.

Definition 1. *The rank distance between two strings u and v is given by:*

$$\Delta(u,v) = \sum_{x \in u \cap v} |ord(x|u) - ord(x|v)| + \sum_{x \in u \setminus v} ord(x|u) + \sum_{x \in v \setminus u} ord(x|v). \tag{1}$$

Example 1. *Let $w_1 = abbab$ and $w_2 = abbbac$ be two strings. Their corresponding indexed strings will be: $\overline{w_1} = a_1b_1b_2a_2b_3$ and $\overline{w_2} = a_1b_1b_2b_3a_2c_1$, respectively. So, $\Delta(w_1, w_2) = \Delta(\overline{w_1}, \overline{w_2}) = 8$*

Remark 1. *The ad hoc nature of the rank distance resides in the last two summations in* **(4.3)**, *where one compensates for unmatched letters, i.e. indexed letters which appear only in one of the two strings.*

Since **(4.3)** penalizes more heavily unmatched letters in the initial part of strings, and this is not the case in biology, we shall add to this value the value given by applying the rank distance to the reverse strings (mirror images). In Section 2.3, up to a trivial normalization, we shall use the more "balanced" variant of the rank distance which follows:

[2] Rank distance can be extended to (potentially) infinite alphabets, but then its computational complexity increases. This generalization is not needed in biological applications, e.g. in the case of DNA strings.

Definition 2. *The average rank distance for strings u and v is defined as:*

$$\Delta_{av}(u,v) = \frac{\Delta(u,v) + \Delta\big(mi(u), mi(v)\big)}{2} \tag{2}$$

where mi(u) is the mirror image of the string u.

The modified rank distance keeps all the metric properties of the "unbalanced" rank distance (1) as used in computational linguistics.
Deletions and insertions are less worrying in the rank case rather than in the Hamming case: if one incorrectly moves a symbol by, say, one position, the Hamming distance loses any track of it, but rank distance does not, and the mistake is quite light. So, generalizations in the spirit of the edit distance are unavoidable in the Hamming case, even if they are computationally very demanding, while in the rank case we may think of *ad hoc* ways-out, which are computationally convenient.

2.2 The complexity of the rank distance

In the DNA sequence analysis, the strings may be huge (for example, for the human genome, the number of nucleotide bases is around 3×10^9), so it is necessary to have good algorithms (regarding time and space complexity) in order to calculate the rank distance between two sequences.
Even if the computation of the rank distance as based directly on its definition may appear to be quadratic, we shall exhibit below two algorithms which take it back to linear complexity.
In (Dinu and Sgarro, 2006) two time linear algorithms to compute (1)are introduced: the first algorithm is a linear time algorithm and it works with a linear supplementary space; basically, it takes back rank distances to the computation of taxi-distances (cf. footnote 1), which are obviously linear. The second one, directly based on the definition of the rank distance, is a little slower than the first, but its advantage is that it has no need of supplementary space. Even if we refer to the quaternary alphabet as needed for DNA strings, the generalization to arbitrary finite alphabets is straightforward.
We mention a convenient property of the rank distance, which allows one to get to get rid of equal aligned bases, in the sense that equal bases are replaced by a new symbol, a dash say, while unequal bases keep their original ranks:

$$\begin{array}{rl}
 & 1\,2\,3\,4\,5\,6 \qquad\quad 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6 \\
u = & a\ a\ g\ c\ c\ t \Rightarrow \bar{u} = a - g - c - \\
v = & c\ a\ a\ c\ g\ t \qquad \bar{v} = c - a - g -
\end{array}$$

If we denote by $\bar{u}$ and $\bar{v}$ the two modified strings, we have, as easily checked: $\Delta(u,v) = \Delta(\bar{u},\bar{v})$. Clearly, to compute the second distance one can forget about dashes.

Algorithm 1 (linear in time and space)

- We use four arrays A[][],C[][],G[][],T[][] with 2 rows and $\max(|u|,|v|)$ columns (bars denote length).
- In each array we shall memorize the following data: A[1][i] will contain the rank of i-th 'a' in the first string and A[2][i] will contain the rank of i-th 'a' in the second string (it will be 0 if there are no more 'a'-s). Analogously for C[][], G[][] and T[][].

- Finally we compute the sum:

$$\sum_i |A[1][i] - A[2][i]| + \sum_i |C[1][i] - C[2][i]| +$$

$$+ \sum_i |G[1][i] - G[2][i]| + \sum_i |T[1][i] - T[2][i]|$$

Example 2. $\begin{array}{lllllll} & 1 & 2 & 3 & 4 & 5 & 6 \\ \bar{u} = & a & - & g & - & c & - \\ \bar{v} = & c & - & a & - & g & - \end{array}$

$A = \frac{1}{3}, C = \frac{5}{1}, G = \frac{2}{4}$ *So,* $\Delta(u, v) = 2 + 4 + 2 = 8$

Remark 2. *The time complexity*[3] *of algorithm 1 is* $\mathcal{O}(|u| + |v|)$.

Remark 3. *In this algorithm, the elements of the matrixes A, C, G and T are integers. The matrixes have 2 rows and* $max(|u|_a, |v|_a)$, $max(|u|_c, |v|_c)$, $max(|u|_g, |v|_g)$ *and* $max(|u|_t, |v|_t)$ *columns, respectively. The total number of columns of the 4 matrixes is at most equal to* $|u| + |v|$. *So, the supplementary space is* $\mathcal{O}(|u| + |v|)$.

Algorithm 2: (without supplementary space)

- We'll use eight positive variables *ia, ic, ig, it* and *ja, jc, jg, jt* which will point to the last *a, c, g* or *t* read in the first (i) and second (j) string (initially all are 0)
- So, if we read in first string an 'a', we search in the second string the next 'a' starting from the position ja; if it is found, we make the difference $|ja - ia|$ and add it to the final sum. Analogous with c, g and t.

2.3 Experimental data

To test our method in bioinformatics, we use a classical problem: the phylogenetic analysis of the mammals.

We use whole mitochondrial DNA sequence genome of the following 22 mammals available in the EMBL database: human (Homo sapiens, V00662), common chimpanzee (Pan troglodytes, D38116), pigmy chimpanzee (Pan paniscus, D38113), gorilla (Gorilla gorilla, D38114), orangutan (Pongo pygmaeus, D38115), sumatran orangutan (Pongo pygmaeus abelii, X97707), gibbon (Hylobates lar, X99256), horse (Equus caballus, X79547), donkey (Equus asinus, X97337), Indian rhinoceros (Rhinoceros unicornis, X97336), white rhinoceros (Ceratotherium simum, Y07726), harbor seal (Phoca vitulina, X63726), gray seal (Halichoerus grypus, X72004), cat (Felis catus, U20753), fin whale (Balenoptera physalus, X61145), blue whale (Balenoptera musculus, X72204), cow (Bos taurus, V00654), sheep (Ovis aries, AF010406), rat (Rattus norvegicus, X14848), mouse (Mus musculus, V00711), North American opossum (Didelphis virginiana, Z29573), and platypus (Ornithorhyncus anatinus, X83427).

Our approach is the following: for any two mtDNA sequences used in this study, we compute the *normalized average rank distance*. To normalize the average rank distance as in (2), we divide Δ_{av} by the maximum possible value which can be reached by Δ_{av}. The maximum

[3] The so-called *bit complexity* of algorithm 1 is $n \log n$ rather then n, due to the fact that we have to write down longer and longer integer representations, e.g. decimal representations, for the indexes. The algorithmic complexity referred to here is, however, the standard one, since in practice the extra complexity due to integer indexing is neglected in current computational models.

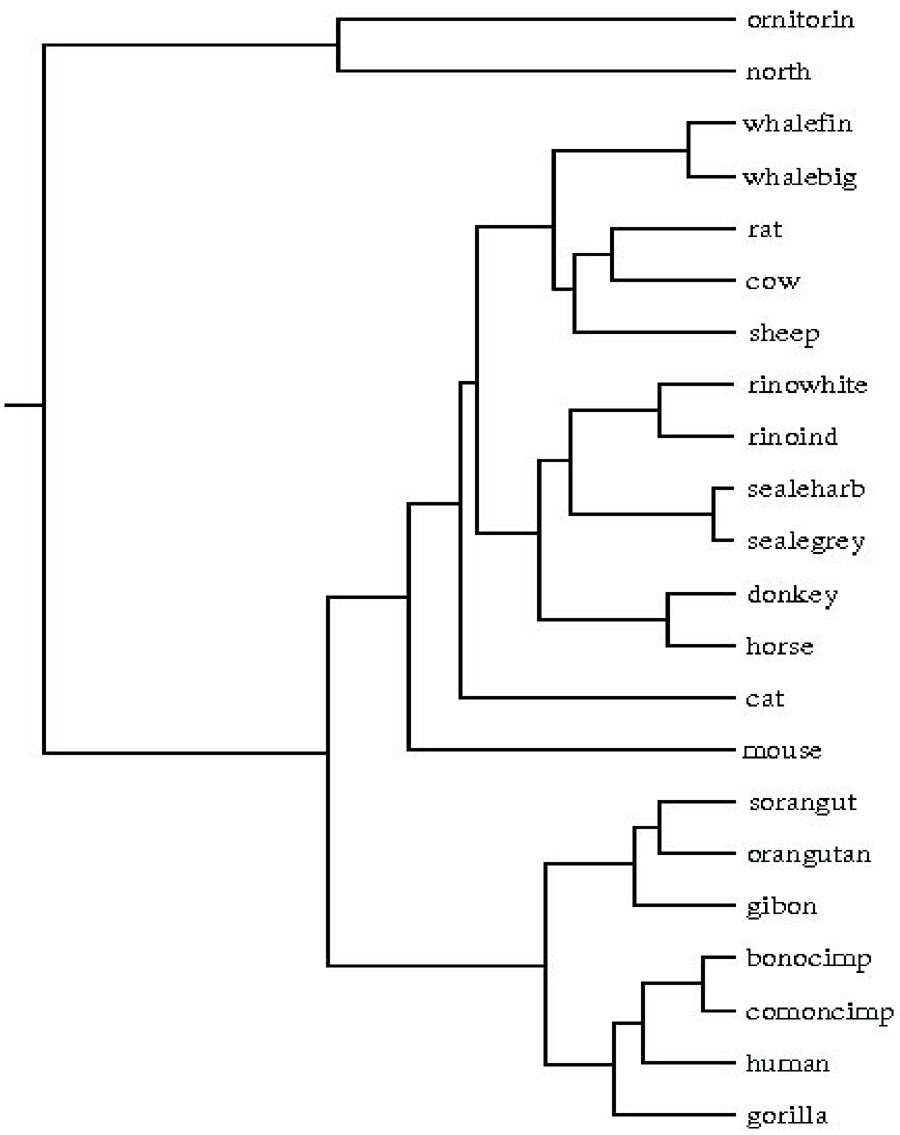


Fig. 1. The mammals phylogenies build from complete mammalian mtDNA sequences using rank distance

value[4] between two strings u and v is equal to $\frac{|u|(|u|+1)+|v|(|v|+1)}{2}$, as soon checked. So, the normalized average rank distance between two strings is:

$$\overline{\Delta}_{av}(u,v) = \frac{\Delta(u,v) + \Delta(mi(u), mi(v))}{|u|(|u|+1) + |v|(|v|+1)} \tag{3}$$

In our experiment, the usual length of a mtDNA sequence is around 2^{14} letters. Using the normalized Δ_{av}, we computed the distance matrixes for all mammal mitochondrial DNAs

[4] Actually, this is the maximum value when the alphabet size is n or more, else it is only an upper bound. Even for "small alphabet sizes", however, the maximum value is quadratic in n: think e.g. of two quaternary sequences $r_a r_b r_c r_d$ and $r_b r_a r_c r_d$, obtained by juxtaposition of four runs of the same letter repeated $n/4$ times, at rank distance $n^2/2$, as soon checked.

reported in the upper experiment[5]. Then we used Neighbor Joining method to construct the phylogenies for the mammals.

In Fig. 1 we have drawn the phylogenetic tree of the mammals obtained with the third extension of our distance.

The tree that we have obtained has a topological structure comparable to the structure of other trees reported by other researches (27), (7), (26). Though, two differences can be observed: the classifications of rat and cat in our research are not similar to its corresponding classifications in the other papers. If we look at the distance matrix, we can observe that the grey seal and harbor seal are the closest mammals to the cat, but the cat is not the closest mammal to the grey seal and the harbor seal. The same with the mouse and the rat (the rat is the closest mammal to the mouse, but the mouse is no the closest mammal to the rat).

3. Circular approaches

A limitation of Rank Distance with respect to genomic applications is more general and is common to other more established distances (like Levenshtein, Hamming, and other): if two sequences have multiple subsequences in common but these subsequences appear in widely different orderings in the two sequences, then Rank Distance has trouble to distinguish the two sequences as being closer than two essentially random sequences.

It is important to note that the limitation is rather prominent for genomic applications because of the fact that the existence and extent of common parts is far more important than the ordering or shifting between the common parts.

To overcome the identified limitation of the classical Rank Distance two basic strategy are introduced in (Dinu and Ghetu, 2011): in computing the distance between two partial rankings, we investigate the rank distances between the smaller sequence and all the circular permutations of the larger sequence. We have two natural ways to do this: either sum all distances, either take the minimum one between all the circular permutations.

One of the drawbacks of using the classical Rank Distance in genomic applications is that it needs some additional balancing mechanism. The rank distance calculation heavily penalizes unmatched elements at the end of the sequences and this has no biological significance or rationale. Because of this, the rank distance is usually applied to genomic problems in its balanced form where one calculates both the normal rank distance and the rank distance on the reverse sequences. Circular rank distance does not suffer from this type of problem since it is naturally balanced by the circular permutations involved. Calculating the distance between the reverse strings brings no real benefit.

3.1 Total circular rank distance on strings

This approach promises to better reward similarity between the sequences because each pair of similar subsequences will have the opportunity to be better aligned in some suitable Rank Distance computations (between the smaller ranking and some circular permutations of the larger one). These favorable alignments should serve to markedly decrease the overall Circular Rank Distance score below the score of random rankings. We should expect the distance decrease to be proportional to the extent of the similarity between subsequences of the two rankings

[5] Two more ad-hoc extensions of the rank distance (cf. remark 1) were tested in our experiments: in both one prolonged the strings so as to take them to the same length, either by adding "tails" or by random insertions. Results do not show any significant variation, which further validates the *robustness* of the rank distance.

Although it is clear that what we defined as the Circular Rank Distance is of no use for measuring distances between rankings, the same idea can be adapted to measuring distances between strings. With strings, there can be repeating elements/characters in each sequence and with genomic strings in particular (nucleotides or amino-acids sequences) we are dealing with very long strings over a very small alphabet (4 or 20 elements in size). This is particularly advantageous to the Circular Rank Distance approach as it completely removes the problem of irrelevance for the larger sequence and it brings back the advantages we initially sought: it rewards commonality between the sequences, irrespective of the ordering of the similar parts or their relative positions in the two sequences.

Definition 3. *We define the Circular Rank Distance between two strings S_1 and S_2 of lengths N_1 and N_2 as being the sum of the rank distances between S_1 and each circular permutation of S_2.*

It is important to note that each circular permutation of S_2 will have to have its characters renumbered for calculating its rank distance to S_1.

Example 3. *CRD("AACGTT", "TTGCAA") = RD("AACGTT", "TTGCAA") + RD("AACGTT", "TGCAAT") + RD("AACGTT", "GCAATT") + RD("AACGTT", "CAATTG") + RD("AACGTT", "AATTGC") + RD("AACGTT", "ATTGCA") = 18 + 12 + 8 + 8 + 8 + 12 = 66.*

In this example, if we average the Circular Rank Distance value to the number of rank distances it summed, we obtain a value of 11 which is significantly lower than the value of 18 that would have been obtained with the classical Rank Distance. This is because the circular rank distance rewards the fact that the two input sequences have common substrings "AA" and "TT", while the classical Rank Distance cannot distinguish this situation.

3.2 Minumum circular rank distance

Another variation of the Circular Rank Distance on strings , that can be considered is to select the minimum Rank Distance for all the circular permutations considered. We can define the Minimum Circular Rank Distance to be:

Definition 4. *Minimum Circular Rank Distance between two strings S_1 and S_2 of lengths N_1 and N_2 ($N_1 \leq N2$) is:*

$$MCRS(S_1, S_2) = \min_{1 \leq i \leq N2} RD(S_1, CP(S_2, i))$$

, where $CP(X, i)$ is the ith circular permutation of string X.

Using this variant of the circular Rank Distance we are choosing the distance to be the score of the best alignment between the smaller string and a circular permutation of the larger one. This approach has the advantage of aggressively rewarding similarity between the sequences but has the drawback of potentially being overoptimistic. Furthermore, when evaluating pair wise distances in a set of strings, a more conservative distance estimate, like the normal Circular Rank Distance, could be preferable for offering more consistently correlated distance estimates across all pairs.

3.3 Experiments

In order to compare distances between two pairs of strings it is necessary to judge them on the same scale. It is of no use to directly compare the distance between two rankings of small length with the distance between two rankings of sizeable length, as the two values will have different orders of magnitude. It would be more useful to scale these values to some common

interval first. This is achieved by normalizing each distance to the same interval. In order to normalize the circular rank distance to interval [0, 1] we need to divide it by the maximum rank distance that is possible. The maximum circular rank distance between two strings of length N_1 and N_2 (with $N_1 < N_2$) is achieved when the two strings have no common element.
In order to test the effectiveness of the Circular Rank Distance with real data, we took to construct a phylogenetic tree of some principal mammalian representatives by comparing their mitochondrial DNA (mtDNA) downloaded from the EMBL database. We chose 21 mammals, with the average size of their mtDNA sequences being approximately 16500 nucleotides each. We first computed the circular rank distance between each pair of species, normalized to [0, 1], then fed the distance matrix to a standard Neighbor Join algorithm. We ran the most straightforward implementation of the Circular Rank Distance and the Minimum Circular Rank Distance with no special optimizations and no parallelization. The most intensive part of the process, computing the distance matrix, took approximately 27 minutes for each distance type, on an Intel Core2 Duo 2.5GHz processor with plenty of memory.

The dendrograms are represented in figures 2 and 3.

As it can be seen from the phylogenetic trees that were produced, the Circular Rank Distance and the Minimum Circular Rank Distance allowed a good and very similar representation of the sample species. With the odd exception of the donkey and horse, all the other species have fairly common-sense positions and are correctly grouped into primates, sea/water mammals and rodents.

4. Distances and distinguishabilities

Intuitively, *distance* is a measure of how close or far apart two physical objects or concepts are. In many situations the distance is given in terms of length, time delay, rank difference, etc. An alternative measure, *(di-)similarity*, is sometimes used and in many cases the two measures are related. In (10) the authors attempt to classify, based on specific areas of interest, the huge number of distances currently defined and used in various studies.

Rank distance (including its circular approaches too), as most tools of this kind, provides an evaluation which is strictly limited to the closeness of two objects in themselves, without looking at the relation of the respective objects with the rest of universe. In a way, the rank distance is local, as most distances are, while we looking a global non-local variant thereof.

From a mathematical point of view, a distance measure provides an evaluation mostly limited to the two objects themselves, with the Triangle Inequality as the only influence of the rest of the universe. For disimilarities, even that connection is removed.

However, there are a lot of situations where the closeness between two objects is significantly influenced by the presence or absence of other objects in the universe.

While not neglecting the intrinsic quantitative measure between two objects, the main concept discussed in this section, *distinguishability*, has the following motivation: two objects are as close as the similarity of their behavior with respect to all the other objects of the universe considered.

For example, when a new DNA sequence is given, the first step taken by a biologist is to compare the new sequence with sequences that are already well studied and annotated. Sequences that are similar would probably have the same function.

We emphasize this point of view in the present section: even though the absolute distance may be large, it may look relatively small when the two objects are situated in a sparse area of the universe, with many holes and few neighbours. We make use of the notion of distinguishability between two sequences (basically a Shannon-theoretic notion closely related to the problem of the median) and we investigate it in the metric space of rank

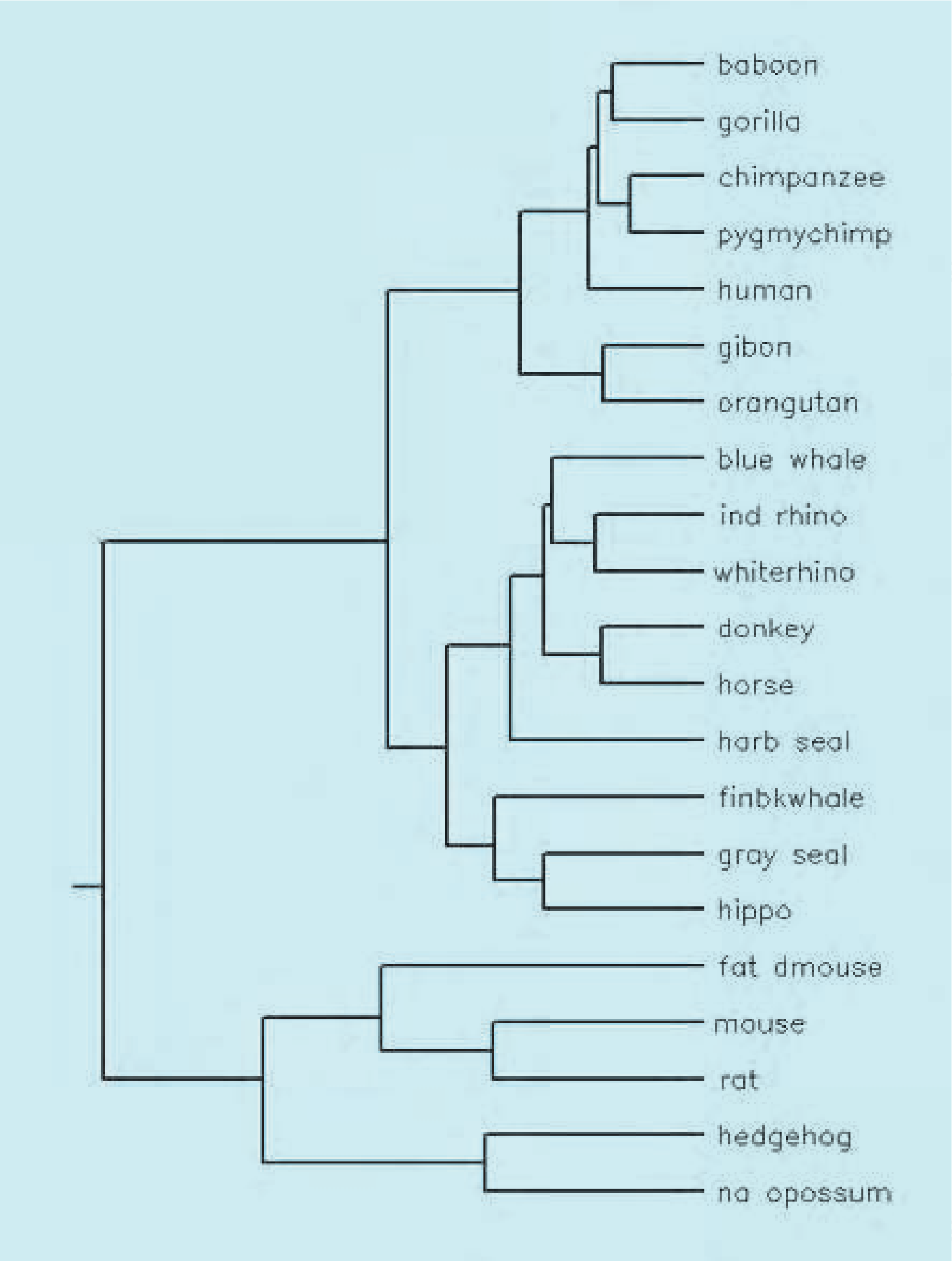


Fig. 2. The mammals phylogenies build from complete mammalian mtDNA sequences using CRD

distances, thinking of application in biology, first of all in DNA word design, a form of coding for biological computers where the codewords are DNA strings.

4.1 Foreword

We consider a noisy channel where letters can be transposed but not replaced; we decode by minimising the Spearman footrule distance between input and output. Footrule coding serves to stress the difference between codeword distance and codeword distinguishability, two notions confused to no harm in the case of standard codes based on Hamming distances, and yet definitely apart from each other; we show that discriminating distance

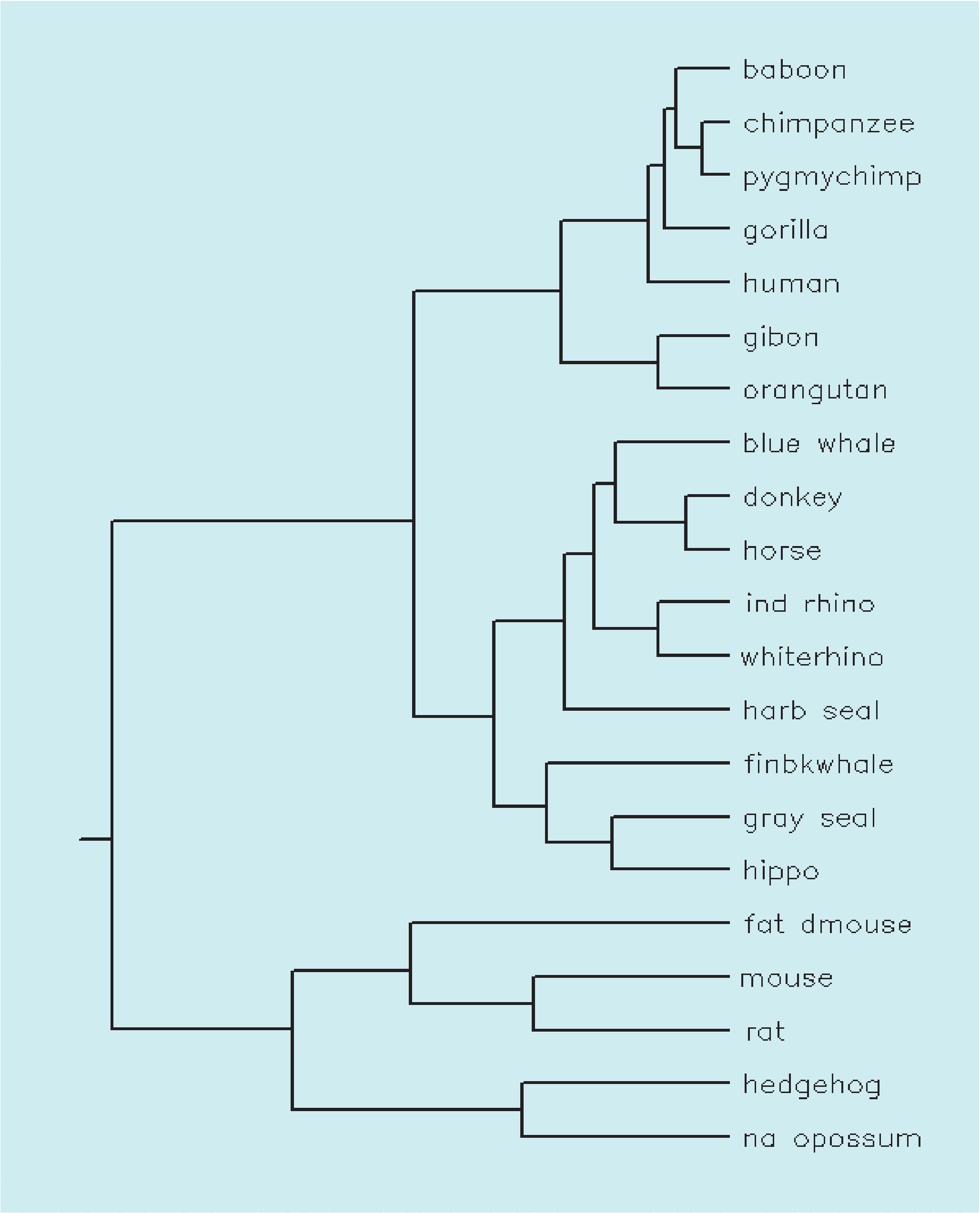


Fig. 3. The mammals phylogenies build from complete mammalian mtDNA sequences using MCRD

from distinguishability is essential when one moves to realistic codes which correct both replacements and tranpositions.

Inadvertently transposing two letters is quite a common source of error when typing, but standard error-correcting codes based on the Hamming-distance between strings envisage only letter substitution (replacements) and so end up over-estimating letter transpositions: e.g. a single *twiddle* (a single transposition between adjacent positions) counts as much as *two* substitutions. One might make use of other string distances which assign a smaller "penalty" to twiddles, e.g. suitable variants of the edit distance as in Section 4.5: in building these and other unusual codes, however, special care should be taken to discriminate between two distinct notions which in standard coding are confused to no special harm, *codeword distance* and *codeword distinguishability*. In this correspondence we consider a model of channel noise which envisages *only* letter transpositions: one decodes by minimising the rank distance (or Spearman footrule, when work with full rankings) distance between input and output. Unrealistic as it is, the model serves as a warning not to be overlooked when constructing error-correcting codes: in the final Section 4.5 we argue that the "odd" case of footrule coding appears to be quite typical of realistic situations when channel noise includes both replacements and transpositions.

4.2 Rank distinguishability

If d is a non-negative "distance" between the elements[6] of a finite set $\mathcal{X}$, the *distinguishability* between any two such elements x and y is defined as:

$$\delta(x,y) = \min_{z \in \mathcal{X}} \max \left[d(x,z), d(y,z) \right] \tag{1}$$

and so it is achieved by any *geometric (combinatorial) centre* z of the set $\{x,y\}$.

If d happens to be an *integer metric distance*, as we shall always assume[7] below, one soon proves:

$$\left\lceil \frac{d(x,y)}{2} \right\rceil \leq \delta(x,y) \leq d(x,y) \tag{2}$$

The left-hand bound holds uniformly for Hamming distances, while the right-hand bound holds uniformly iff d is a ultrametric; for an ultrametric string example take the distance between strings of the same length equal to the first position i, left to right, where the two strings differ. For "unruly" metric distances one may have "unruly" behaviors: e.g. take the integers $\{0,4,7,10\}$ with Euclidean distance: 0 and 4 achieve the upper bound $\delta(0,4) = d(0,4) = 4$, while 4 and 10 achieve the lower bound $\delta(4,10) = d(4,10)/2 = 3$. As for 0 and 10 their distinguishability is 6, while the bounds are 5 and 10, respectively, and so none of the two bounds is achieved. Observe that $\delta(4,10) < \delta(0,4)$ even if $d(4,10) > d(0,4)$.

Rather than this toy example, below we shall take into account the string geometry based on a metric distance between strings of length n over a letter alphabet $\mathcal{A}$ of size q, namelly *rank distance*. The idea of resorting to an *ordinal* distance, is that a string is received as such across the channel, only the *order* in which its letters are output may have been scrambled by channel noise. With respect to other ordinal distances, the rank distance is attractive because of its low complexity, linear in the string length n as Hamming distances: fair to say, however, we chose it because, at least for $q \geq 3$, cf. Section 4.3 and the example in Section 4.4, it gives rise to an "odd" string geometry which goves us the chance to stress the basic difference between codeword distance and codeword distinguishability.

4.3 Footrule distinguishability

Say $x = x_1 x_2 \dots x_n$ is a string of length n in the composition[8] *class* $\mathcal{K} \subset \mathcal{A}^n$. We may *index* any such string in the following way: if a letter a occurs $m > 0$ times, in the indexed version of x

[6] The distance might as well be a *distortion* between elements of an input set $\mathcal{X}$ and those of a distinct output set $\mathcal{Z}$, where z as in (2) would be constrained to belong; the only requirement would be that $\forall x \; \exists z : d(x,z) = 0$. Even if $\mathcal{X} \neq \mathcal{Z}$, however, the distinguishability, unlike the distance, involves two *input* elements.

[7] We shall further assume that the distances actually taken on are *consecutive* integers; for arbitrary distances, the integer ceiling in (2) should be understood as the lowest distance $\geq d/2$.

[8] A *composition class* is made up of a string together with all of its permutations.

we write those m occurrences as $a_1, a_2, \ldots, a_m$ in the order as they appear; e.g. $x = aabbca$ is indexed to $a_1a_2b_1b_2c_1a_3$. If the j-th component in x is the (indexed) letter a_i, we define the *rank* $\pi(a_i|x)$ of a_i in x to be its position j; e.g. in our example $\pi(b_2|x) = 4$.

Definition 5. *(Spearman footrule distance, or rank distance)*

$$d(x,y) \;=\; \frac{1}{2}\sum_{a}\sum_{1\le i\le n_a} |\pi(a_i|x) - \pi(a_i|y)| \;\;, \quad x, y \in \mathcal{K}$$

(A void summation contributes 0) Observe that the number of occurrences n_a is the same in any string of the composition class $\mathcal{K}$. Footrule distinguishabilities are bound to be integers, as footrule distances are; footrule distance 1 corresponds to a single twiddle between adjacent positions (a single twiddle is a good "unit" for ordinal distances: that is why we prefer to divide by 2 in definition 1). We need the following result:

Theorem 1. *(19) The string z lies on the closed segment $[x, y]$ iff either $\pi(a_i|x) = \pi(a_i|z)$ or $\pi(a_i|z) = \pi(a_i|y)$ or:*

$$\mathrm{sign}\big(\pi(a_i|x) - \pi(a_i|z)\big) = \mathrm{sign}\big(\pi(a_i|z) - \pi(a_i|y)\big) \quad \forall a_i$$

We are ready to prove a result which is quite obvious in a Euclidean or in a Hamming geometry:

Theorem 2. *(4) The minimum in (1) is achieved by at least a point z lying on the closed segment $[x, y]$*

Proof. If three sequences x, z, y violate the collinearity condition in Theorem 1 there must be an occurrence a_i with $\max\left[\pi(a_i|x), \pi(a_i|y)\right] < \pi(a_i|z)$, or an occurrence b_j with $\min\left[\pi(b_j|x), \pi(b_j|y)\right] > \pi(b_j|z)$. We find it convenient to say that in these two cases there is an "arrow" pointing forward, or an "arrow" pointing backward, respectively; if the arrow points forward we shall say that its "depth" is $\pi(a_i|z)$, if it points backward its depth is $(n+1) - \pi(b_j|z)$. Assume that the set $\mathcal{M}$ of minimizing z's has void intersection with the segment $[x, y]$: then each z in $\mathcal{M}$ has at least an arrow. ... □

As an example, take x =123 and y =321; distance and distinguishability are equal to 2 and so the right-hand bound holds with equality: the closed segment is made up just of its two extremes[9] x and y, but also the string z =213 outside the segment, being at distance 2 from x and y, achieves the minimum. To get an example where the open segment is not void, prolong x and y to $x' = x1212$ and $y' = y2121$, respectively, and add the suffix 1221 to get the three minimising strings $x1221$, $y1221$ on the segment and $z1221$ outside the segment ($n = 7$). In this case, $d(x', y') = 4$, $\delta(x', y') = 3$, and so none of the two bounds in (2) holds with equality. The footrule distance is often compared to the *twiddle distance*, called also *Kendall τ-distance* [], which simply counts the minimum number of exchanges between consecutive positions needed to transform x into y; now, take $x = 1234$, $y = 4321$: as soon checked, 2341 is centre for both Kendall and Spearman distance achieving the lower bound (2) in both cases, it lies on the Kendall segment but does not lie on the Spearman segment; instead, 1324 is a footrule centre which lies outside both segments, but it is not a Kendall centre (one soon proves that Kendall centres lie all on the Kendall segment and uniformly achieve the lower bound (2)).

[9] The right-hand bound in (2) is achieved with equality iff the open segment $]x, y[$ is void, i.e. when there is no z other than x and y for which the triangle inequality $d(x,z) + d(z,y) \ge d(x,y)$ holds with equality.

Computer simulations have shown that "many" footrule centres lie outside the corresponding footrule segment.

It is no coincidence that these "geometrically nasty" examples are ternary or quaternary. Instead, *binary* strings on $\{0,1\}$, are definitely more friendly and "Hamming-like" than in the case $q \geq 3$:

Theorem 3. *(Binary footrule distinguishability) For two binary strings the lower bound in (2) is always achieved.*

Proof. It was shown in (18) that in the binary case the footrule distance is equal to the Kendall τ-distance, cf. above (to show this equality, just use an induction on the twiddle distance). Now, proving that, whatever the alphabet size q, the twiddle distinguishability achieves the lower bound in (2) is straightforward (think of the suboptimal bubble algorithm for sorting, which uses precisely twiddles). The expression on the right holds because 0's and 1's give the same contribution to the binary footrule distance, as proved in (18) (clearly, one may use 0 instead of 1 in the formula above). □

Algorithmically, the impact of Theorem 2 is limited, since there are over-populated segments[10] already in the binary case; take e.g. $x = 0^m1^m$, $y = 1^m0^m$, $n = 2m$: *all* the exponentially many strings of Hamming weight m lie on the segment $[x, y]$.

4.4 Footrule codes

For footrule-distance codes one should *not* adopt the following procedure, misleadingly suggested by what one may do with Hamming distances:

1. Fix a threshold τ; form the *codebook* $\mathcal{C}$ by taking as many input strings as possible at pairwise distance $\geq \tau$
2. Decode the output string z to a codeword x which minimises the distance $d(x,z)$

All works, however, if the word *distance* in point 1, but *not* in point 2, is replaced by the word *distinguishability*. This ensures the following:

Reliability criterion: *If the codeword x is fed to the channel, and if the distance $d(x,z)$ to the observed output z is strictly less than τ decoding is successful; conversely, there is at least a couple x, z ... which brings about a decoding error*.

Note that what we are doing, in particular the reliability criterion, works *whenever* we have a distance between inputs and outputs, and whenever we decode as in point 2, even when the input and the output set are distinct. In the case of Hamming distances and binary footrule distances, the distinguishabilty $\delta(x,y)$ achieves uniformly the lower bound, and so it turns out to be "almost" the same as half the distance $d(x,y)/2$: one can get along quite well forgetting entirely about distinguishabilities; actually, the fact that the monotone dependence of δ on d is only weak can be put to good use in *error detection*, cf. below this Section. It can be shown that, whenever the lower bound is uniformly achieved, everything, error correction and error detection, works exactly as in the usual Hamming case, so trivialising the opposition distance

[10] A software tool to compute (short) footrule segments is available at http://rankd.info; quicker algorithms to compute directly a centre on the segment are badly needed. We recall that the problem of finding footrule centres for an *arbitrary* number of permuted strings is NP-hard.

vs. distinguishability. However, this is not so in general, as the case of footrule codes on alphabet with size $q \geq 3$ clearly shows.
We move to error detection. The basic notion is that of an *even couple*.

Definition 6. *An* even *couple x, y is one for which any centre as in (2) is at equal distance from x and y.*

The following theorem holds in any integer metric space where at least a centre lies on whatever segment, as is our case.

Theorem 4. *The couple x, y is an even couple iff $\delta(x, y) = d(x, y)/2$, which implies that the distance $d(x, y)$ must be an even integer; in general, the latter condition is not sufficient to have an even couple.*

Proof. 1234 and 4321 are at even distance 4; their distinguishability is 3 and a centre is 1324 at distances 1 and 3 from the two extremes; a centre outside the segment is 2413, at equal distance 3 from both extremes. □

An open problems for footrule distances is finding a simple criterion to understand when a couple is even.

4.5 History and perspectives

The notion of codeword distinguishability, or, symmetrically and equivalently, of *codeword confusability*, goes back to Cl. Shannon, and more precisely to his *zero-error information theory*, dated 1956, cf. (30). In Shannon's case the distance between input and output is 0 when the probability of the transition is positive even if "small", else it is 1; the distinguishability function is 0 or 1, according whether there is, or there is not, at least one output which is accessible from both codewords with positive probability. The second author has put forward a multi-step generalisation of zero-error information theory, called *possibilistic*[11] *information theory*, in which the distance (dissimilarity) between input and output is no longer constrained to have only two distinct values: distinguishability in the multi-step theory is exactly the notion defined in Definition 1. The possibilistic framework can be shown to be ample enough to accommodate not only Hamming-distance codes but also "odd" forms of coding as ours here or those found in DNA word design, where codewords are DNA strings.

5. Conclusions and future works

In this chapter we have exhibited a low-complexity distance for DNA sequence comparison. We showed that our distance can be computed in linear time. Another advantage of our method is that it imposes minimal hardware demands: it runs in optimal conditions on modest computers, reducing the costs and increasing the number of possible users. Our experiments on the phylogenies of mammals produced results which are quite similar to those reported in the literature, but whose computational costs in time and space were definitely lower. This computational gain may turn out to be quite useful in situations where the use of ŞexactŤ methods leads to computational stumbling blocks, as often happens in biological applications.
We proposed two variants of rank distance: circular and minimum circular rank distance. As it can be seen from the phylogenetic trees that were produced, the Circular Rank Distance and

[11] The name derives from the fact that the multi-step theory can be obtained from probabilistic information theory if one replaces probabilistic notions by the corresponding possibilistic ones, possibiliy theory being a form of multi-valued logic, or also of interval-valued probability.

the Minimum Circular Rank Distance allowed a good and very similar representation of the sample species. With the odd exception of the donkey and horse, all the other species have fairly common-sense positions and are correctly grouped into primates, sea/water mammals and rodents.

The main advantage of using the circular rank distance is its ability to reward two strings containing (any number of) similar subsequences with a lower distance irrespective of the order or place in which the similar subsequences appear in the strings. The main disadvantage is the quadratic time complexity it requires for being computed (in the direct, straightforward manner).

This chapter is only introducing a new and promising Rank Distance variation for bioinformatics applications. There is still much work to be done in exploring the subject, with some of the most important directions for further research being:

- Exploration of the mathematical properties of the Circular Rank Distance. Is the new distance a proper metric? Exploration of the analytical or statistical notions of medians or centers for it.
- Finding a more efficient computation of the Circular Rank Distance. There are potent particularities that can be exploited to achieve a sub-quadratic time complexity in either asymptotic or amortized measuring

6. References

[1] L. Bortolussi, A. Sgarro. Noise of DNA word design is not stochastic. *submitted*

[2] L. Bortolussi, A. Sgarro. Possibilistic channels for DNA word design. in *Soft Methods for Integrated Uncertainty Modelling*, ed. by J. Lawry et alt., Advances in Soft Computing, Springer Verlag (2006), pp.327-335 .

[3] L. Bortolussi, L.P. Dinu, A. Sgarro. Codeword distinguishability vs. codeword distance: the case of rank coding. *submitted*

[4] L. Bortolussi, L.P. Dinu, A. Sgarro. Geometry of strings and possibilistic coding. In Proc. ASMDA 2011, Rome, Italy (to appear)

[5] A. Condon, R.M. Corn, and A. Marathe. On combinatorial dna word design. *Journal of Computational Biology*, 8(3):201–220, November 2001.

[6] Fr.P. Brooks jr.˙ Three great challanges for half-century-old Computer Science. *Jrnl of the ACM*, Vol.50, No.1, Jan. 2003, pp. 25-26.

[7] Y. Cao, A. Janke, P. J. Waddell, M. Westerman, O. Takenaka, S. Murata, N. Okada, S. Paabo, M. Hasegawa, *Conflict among individual mitochondrial proteins in resolving the phylogeny of Eutherian orders*, J. Mol. Evol., 47(1998), 307-322.

[8] Th.H. Cormen, Ch.E. Leiserson, R. Rivest and Cl. Stein. *Introduction to Algorithms*, 2nd ed., MIT Press and McGraw Hill, 2001

[9] I. Csiszár and J. Körner. *Information Theory*, Akadémiai Kiadó (Budapest) and Academic Press (New York), 1981

[10] E. Deza, M.M. Deza. *Dictionary of Distances*. Elsevier, Amsterdam, 2006.

[11] P. Diaconis, R.L. Graham, Spearman Footrule as a Measure of Disarray, *Journal of Royal Statistical Society. Series B Methodological*, 39(2), 262-268, 1977

[12] L.P. Dinu, 2003. On the Classification and Aggregation of Hierarchies with Different Constitutive Elements. *Fundamenta Informaticae*, 55(1), 39-50, 2003

[13] L.P. Dinu. Rank Distance with Applications in Similarity of Natural Languages. *Fundamenta Informaticae*, 64(1-4), 135-149, 2005

[14] A. Dinu, L.P. Dinu. On the Syllabic Similarities of Romance Languages. In *A. Gelbukh (Ed.): CICLing 2005. LNCS 3406*, Lecture Notes in Computer Science vol. 3406, 785-788, 2005

[15] L.P. Dinu, F. Ghetu. Circular Rank Distance: A new approach for genomic applications. In Proc. BioKDD 2011, Toulouse, France, 2011 (to appear)

[16] L.P.Dinu and F. Manea. An Efficient Approach for the Rank Aggregation Problem. *Theoretical Computer Science*, vol. 359 (1-3), pp. 455-461, 2006

[17] L.P. Dinu, A. Sgarro. A Low-complexity Distance for DNA Strings. *Fundamenta Informaticae*, 73(3), 361-372, 2006

[18] L.P. Dinu, A. Sgarro. Maximum rank distance for binary strings. *Mathematica Pannonica*, 19/1, 125-133, 2008.

[19] Dinu, L. P. and Ioan Tomescu, 2009. *From rankings' collinearity to counting SDR via chromatic list expression*, Int. J. of Computing Mathematics, 86(9), 1483-1489

[20] L.P. Dinu, Marius Popescu and Anca Dinu. 2008. Authorship Identification of Romanian Texts with Controversial Paternity. *Proceedings 6-th LREC 2008*, Marrakech, Morocco

[21] D. Dubois and H. Prade, eds. *Fundamentals of Fuzzy Sets*. Kluwer, 2000.

[22] M.R. Garey and D.S. Johnson. *Computers and Intractability*, W. H. Freeman and Company, New York, 2000

[23] C. de la Higuera and F. Casacuberta, Topology of Strings: Median String is NP-complete, *Theoretical Computer Science*, vol. 230 (39-48), 2000.

[24] J. Körner, A. Orlitsky. Zero-error information theory. *IEEE Trans. Inform. Th.*, 44 (6) pp. 2207-2229 (1998).

[25] E.V. Koonin. The emerging paradigm and open problems in comparative genomics. Bioinformatics, 15, 265-266, 1999

[26] M. Li, X. Chen, X. Li, B. Ma, and Paul M.B. Vitanyi. The Similarity Metric. IEEE Transaction on Information Theory, 50:12(2004), 3250-3264, 2004.

[27] A. Reyes, C. Gissi, G. Pesole, F. M. Catzeflis, and C. Saccone *Where Do Rodents Fit? Evidence from the Complete Mitochondrial Genome of Sciurus vulgaris* . Mol. Biol. Evol. 17(6):979 Ű 983. 2000

[28] A. Sgarro. Possibilistic information theory: a coding-theoretic approach. *Fuzzy Sets and Systems*, 132-1, 11–32, 2002.

[29] A. Sgarro and L. Bortolussi. Codeword distinguishability in minimum diversity decoding. *J. of Discrete Mathematical Sciences & Cryptography*, (2006) Vol.9, N.3, pp. 487-502.

[30] C.E. Shannon. The zero-error capacity of a noisy channel. IRE Transactions on Information Theory. Vol. 2 (1956), pp. 8-19.

[31] J. van Lint. *Introduction to Coding Theory*. Springer Verlag, Berlin, 1999.

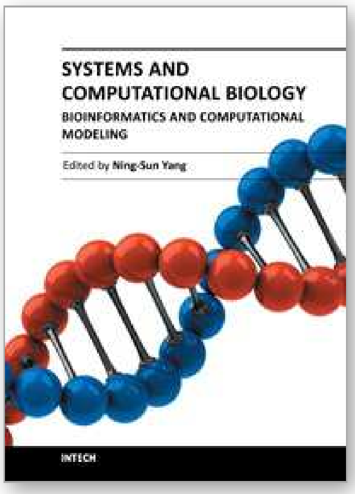
[32] X. Tang, Yang, C.C, 2010. Identifiyng influential users in an online healthcare social network. In Proc. IEEE Int. Conf. on Intelligence and Security Informatics,2010 (ISI '10), May 23-26, 2010, Vancouver, 43-4

[33] J.C. Wooley. Trends in computational biology: a summary based on a RECOMB plenary lecture. J. Comput. Biology, 6, 459-474, 1999

[34] http://rankd.danet.ro

[35] http://www.ebi.ac.uk/embl/index.html

[36] http://evolution.genetics.washington.edu/phylip.html



Systems and Computational Biology - Bioinformatics and Computational Modeling
Edited by Prof. Ning-Sun Yang

ISBN 978-953-307-875-5
Hard cover, 334 pages
Publisher InTech
Published online 12, September, 2011
Published in print edition September, 2011

Whereas some â€œmicroarrayâ€ or â€œbioinformaticsâ€ scientists among us may have been criticized as doing â€œcataloging researchâ€, the majority of us believe that we are sincerely exploring new scientific and technological systems to benefit human health, human food and animal feed production, and environmental protections. Indeed, we are humbled by the complexity, extent and beauty of cross-talks in various biological systems; on the other hand, we are becoming more educated and are able to start addressing honestly and skillfully the various important issues concerning translational medicine, global agriculture, and the environment. The two volumes of this book present a series of high-quality research or review articles in a timely fashion to this emerging research field of our scientific community.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821