Journal of
Data Mining in Genomics & Proteomics

Bolshoy et al., J Data Mining Genomics Proteomics 2014, 5:1
DOI: 10.4172/2153-0602.1000151

Research Article Open Access

Ranking of Prokaryotic Genomes Based on Maximization of Sortedness of Gene Lengths

Bolshoy A[1]*, Salih B[1,2], Cohen I[1,3] and Tatarinova T[4]

[1]Department of Evolutionary and Environmental Biology and Institute of Evolution, University of Haifa, Israel
[2]Department of Computer Science, University of Haifa, Israel, Israel
[3]The Tauber Bioinformatics Research Center at University of Haifa
[4]Children's Hospital Los Angeles, University of Southern California, Los Angeles, California, USA

Abstract

How variations of gene lengths (some genes become longer than their predecessors, while other genes become shorter and the sizes of these factions are randomly different from organism to organism) depend on organismal evolution and adaptation is still an open question. We propose to rank the genomes according to lengths of their genes, and then find association between the genome rank and variousproperties, such as growth temperature, nucleotide composition, and pathogenicity. This approach reveals evolutionary driving factors. The main purpose of this study is to test effectiveness and robustness of several ranking methods. The selected method of evaluation is measuring of overall sortedness of the data. We have demonstrated that all considered methods give consistent results and Bubble Sort and Simulated Annealing achieve the highest sortedness. Also, Bubble Sort is considerably faster than the Simulated Annealing method.

Keywords: Combinatorial optimization; Simulation Annealing; LOPI; Gene length; Computational biology; Evolution

Abbreviations:

1. LOPI: Local Pairwise Interchange
2. B-sort: Bubble Sort
3. COG: Clusters of Orthologous Groups
4. DISC: number of discordant pairs
5. LIS: longest increasing subsequence
6. ED: minimum number of single item deletions
7. MCM: Monte Carlo method
8. SAP: Simulated Annealing Procedure

Introduction

It is natural to group homologous genes into well-defined categories, into gene families. There are several existing approaches. The most popular collection is the database of Clusters of Orthologous Groups (COG) of proteins, containing a comprehensive collection of prokaryotic gene families in complete genomes. Generally, proteins belonging to the same functional family have high sequence similarity; however, their lengths may be substantially different [1-3]. It is not clear how variations of gene lengths (some genes become longer than their predecessors, while other genes become shorter and the sizes of these factions are randomly different from organism to organism) depend on organismal evolution and adaptation. To answer this question we propose to rank genomes according to lengths of their genes, and calculate coefficients of association between genome rank and genome property. The main purpose of this study is to find a computationally effective ranking method that consistently gives reasonable results.

In this work we analyze four methods: Average ranking, Simple Additive Ranking, Bubble Sort and Simulated Annealing. It appearsthat the four selected methods give similar results applied to the same genomic set. Comparing the sortedness obtained by application of the four ranking methods to the matrix of gene lengths, we have found that two Monte Carlo heuristics (Bubble Sort and Simulated Annealing) show the best values of sortedness. Therefore, Bubble sort is a preferable method because it is considerably faster than the Simulated Annealing method.

This paper is organized as follows. Section 2 describes the formal definition of genome-ranking problem, general description of ranking methods, and structure of input data. It also introduces terminology used throughout this paper. Section3presents a detailed description of the selected ranking algorithms, and genomic data as an input to them. Section 4 presents the results and discussion of them. Finally, Section 5presents our conclusions.

Problem Formulation

Ranking

The manuscript is motivated by the following problem: Given a set of objects, with each object described by a set of attributes, the rationale is to find out whether associations between a group of certain object attributes, which we call "object descriptors" and other attributes, which we call "object properties", are significantly non-random. There are different ways to do this [3-5]. Two main approaches are clustering and ranking.Using measures of similarity between the objects based on similarity between the sets of attribute values (object descriptors) one can cluster the objects(for example, genome-classification related applications in [5]), and after that apply methods of multifactor analysis to reveal which object properties are associated with different clusters. We explored this approach, taking genomes as objects and

***Corresponding author:** Bolshoy A, Department of Evolutionary and Environmental Biology and Institute of Evolution, University of Haifa, Israel, Tel: +97258728025 or 16043538414; E-mail: bolshoy@research.haifa.ac.il

Received January 17, 2014; **Accepted** February 10, 2014; **Published** February 13, 2014

Citation: Bolshoy A, Salih B, Cohen I, Tatarinova T (2014) Ranking of Prokaryotic Genomes Based on Maximization of Sortedness of Gene Lengths. J Data Mining Genomics Proteomics 5: 151. doi:10.4172/2153-0602.1000151

gene lengthsas object descriptors [5-7], and found that performed classification closely related to phylogeny and/or taxonomy of prokaryotes. Alternatively, one can work with the precedence relations between the descriptors to rank the objects for further analysis whether object properties correlate with the ranking [3].

A ranking is a relationship between a set of objects such that, for any two objects, the first is either 'ranked higher than', 'ranked lower than' or 'ranked equal to' the second. In a framework of our approach [5-7], every object is defined by a sparse vector of attribute values. So, a sparse matrix presents a set of objects. Permutations of the rows generate different orderings of the objects. There are different approaches to define the optimal order of rows in the matrix (the best ranking). Here we present two possible attitudes: maximization of the sortedness, measured as a minimum of the number of discordant pairs, and the Kemeny-optimal ranking [8,9]. In both cases,Kendall tau distance, based on the Kendall tau correlation coefficient, is used.

Sortedness

To compare quality of ranking obtained as results of different procedures, it is essential to select an appropriate measure of sortedness. Note, that the ranking procedures produce different permutations of rows of the same matrix. The task is not trivial, since each column has an individual sortedness, and ranks of elements in one column do not necessarily agree with the ranks inother columns.

Let us start with the definition of array sortedness and then expand it to matrices. There are several measures for quantifying the sortedness of an array[10]. Here, we present the three most popular measures: DISC, LIS and ED.

a) The number of inversions in a sequence η, denoted by DISC (η), is defined as the number of pairs of items in η which violate the natural ordering property (number of discordant pairs).

b) Another measure for sortedness is the length of the longest increasing subsequence in η, denoted by LIS (η).

c) The third measure is an edit distance to monotonicity, denoted by ED (η), defined as the minimum number of single item deletions needed to reach a sorted sequence η' (it can be easily verified that in fact ED (η)=n–LIS (η)).

Among the three definitions of sortedness, the LIS and ED are more suitable for data stream applications and are not applicable for our purposes, while DISC is as a naturally appropriate definition of the sortedness of a matrix. A pair of numbers either follows the natural ordering or violates it. To apply the DISC criteriato array to a matrix we first should define the precedence rules for the rows, which may be determined in a few different ways, especially when a matrix is sparse:

1) The relationship between two rows r1 and r2 is determined by simple un-weighted comparisons of the elements common to both rows (both elements are not equal to zero):

$$Sg=\frac{1}{K}\sum_{i=1}^{n} c_i(r_1,r_2) \quad (1)$$

Where $c_i(r_1,r_2)=\text{sign}(r_1^i-r_2^i)$ if, $r_1^i.r_2^i \neq 0$ and K is the amount of the elements common to both rows. In other words, if for the majority of the pairs of the common attributes $r_1^i.r_2^i \neq 0$, the condition $r_1^i>r_2^i$ is true then r_1 precedes r_2; if for the majority of common attributes the condition is true,then r_2 precedes r_1; otherwise r_1 and r_2 are tied.

2) The relationship between two rows (objects) r_1, r_2 is determined by the sign of the sum of differences between non-missing elements of the rows:

$$Sg=\sum_{i=1}^{n}(r_1^i-r_2^i)\forall i : r_1^i.r_2^i \neq 0 \quad (2)$$

Having the definition of the precedence relations, we may count all contradictions of the type "r_1 precedes r_2 but r_1 ranked higher than r_2"' in the given ranking, i.e. the number of *discordant pairs*. An optimal ranking will have a minimal number of discordant pairs of objects.

Kemeny-optimal ranking

A complementary to the sortedness approach is an *optimal rank aggregation* approach. The approach is to combine *k* different complete ranked lists of the same set of n elements into a single ranking, which best describes the preferences expressed in the given k lists. This problem dates back to as early as the late 18th century, when Condorcet and Bordain dependently proposed voting systems for elections with more than two objects [11,12]. There are numerous applications in sports, databases, and statistics[13,14] in which it is necessary to effectively combine rankings from different sources.

In the last decades, rank aggregation has been investigated and defined from a mathematical perspective. In particular, Kemeny [8] proposed a precise criterion for determining the "best" aggregate ranking. Given n objects and k permutations of the objects, $\{\pi_1, \pi_2, \ldots, \pi_k\}$, a Kemeny optimal ranking [8,9] of the objects is the ranking π that minimizes a "sum of distances", $p=\sum_{i=1}^{k} d(\pi,\pi_i)$, where $d(\pi,\pi_k)$ denotes a distance between a "total" ranking π and an "individual" ranking π_k. Usually, either Kendall's **τ** rank-correlation coefficient or Spearman's **ρ** rank-correlation coefficient is used to find a distance. Kendall's **τ** is calculated as the difference between the number of concordant and discordant pairs divided by the total number of pairs. For our purposes, we should mention that a Kemeny optimal ranking minimizes the number of pairwise disagreements with the given k rankings.

It is known that finding a Kemeny optimal ranking is NP-hard [15] and remains NP-hard even when there are only four input lists to aggregate [13]. This motivates the problem of finding a ranking that *approximately* minimizes the number of disagreements with the given input rankings. Several approximation algorithms are currently used [13,16].

Solving of the optimization problem

Kemeny optimal ranking may be formulated in terms of solving an optimization problem using either Kendall's **τ** rank-correlation coefficient or Spearman's **ρ** rank-correlation coefficient. As described above, these coefficients provide measures of the degree of correspondence between two ranking vectors. In particular, they assess how well the natural ordering property of the vectors is preserved.

$X^{\tau}=\max_X[\sum_{k=1}^{K}\sum_{i=1}^{N=1}\sum_{j=i+1}^{N} C_{ij}(\vec{X},\vec{r^k})]$, where given a rating vector $\vec{x}$, is equal to 1, if $(r_{x_i}^k < r_{x_{ij}}^k)$; equal to ½, if $(r_{x_i}^k = r_{x_{ij}}^k)$, and 0–otherwise (3)

$X^{ñ}=\max_X \sum_{k=1}^{K}\sum_{i=1}^{N}(X_i-\overline{X})(r_i^k-\overline{r^k})$, where only $r_i^k \neq 0$ are considered (4)

Comparing two pairs of equations (Eq. 1, Eq. 3) and (Eq. 2, Eq. 2) we can see that searching for the maximal sortednessis equivalent to finding the Kemeny optimal ranking in terms of combinatorial optimization.

For our goals Kendall's τ coefficient is more suitable than the Spearman's coefficient. However, finding a solution that maximizes the Kendall's τ rank-correlation coefficient is a difficult, NP-hard [15], task. Therefore heuristic methods, such as the Monte Carlo method [17] and the Simulated Annealing Procedure [18], are frequently used.

Monte carlo methods

In this section we will make aslightly artificial distinction between the Monte Carlo Method (MCM) and the Simulated Annealing (SA) procedure. The Monte Carlo method was developed in the late 1940s by Stanislaw Ulam while he was working on the nuclear bomb project at Los Alamos. It was named by Nicholas Metropolis after the Monte Carlo Casino [19]. In this study we propose to apply MCM to select the best B-rank ranking.

MCM follows the following steps:

Define a domain of possible inputs.

Generate inputs randomly from a probability distribution over the domain.

Perform a deterministic computation on the inputs.

Aggregate the results.

The Metropolis–Hastings algorithm is a Monte Carlo method for obtaining a sequence of random samples from a probability distribution for which direct sampling is difficult or impossible. The algorithm was named after Nicholas Metropolis, who was the first author of [18], and W. K. Hastings, who extended it to the more general case in 1970 [20]. The Simulated Annealing procedure is an adaptation of the Metropolis-Hastings algorithm. The method was described by [21] and by [22]. This method simulates behavior of a physical system, the internal energy of which is to be minimized. The goal of the procedure is to bring the system from an arbitrary initial configuration to a configuration with the minimum possible energy. At each step, SA considers some neighboring configuration ψ' of the current configuration ψ and probabilistically decides between moving the system to configuration ψ' or staying in configuration ψ. These probabilities ultimately lead the system to move to configurations of lower energy.

Gene Length-based model

In the works of Bolshoy et al. [5,6] a "gene length based" model was introduced. It is an algebraic model to represent genomes as vectors of genes. The set of genomes is represented as a matrix, in which each row stands for a genome and each column stands for a gene family, and each item stands for the length of a member of a gene family i in a genome j. In our study, the objects are prokaryotic genomes; the descriptors are the lengths of the genome proteins indexed according to the certain database. (We usethe database named "Clusters of Orthologous Groups of proteins database", see below in Materials and Methods.) Rankings of the selected genomes are performed using the descriptors in order to calculate coefficients of association between a genome rank and a genome property. The descriptors are of a homogenous nature–they're all positive integers. There are different types of genome properties, i.e., a prokaryote is either *Archaea* or *Bacteria*; an organism may be hyper thermophile, thermophile, psychrophile or mesophile; a genome has a certain GC-content, and so on. There are different types of data: Nominal, Ordinal, Interval, and Ratio. There are different types of data of genome properties as well: Kingdom is a nominal dichotomous (binary).

Variable: thermophilicity is an ordinal variable; GC composition is a ratio variable; and number of genes is an interval variable.

Materials and Methods

COGs database

The presented procedures are evaluated on the subset of the database of Clusters of Orthologous Groups of proteins (COGs) [1,2,23]. The principles of the database construction are described by [23]. Briefly, the COGs were constructed by applying the criterion of consistency of genome-specific best hits to the results of an exhaustive comparison of all protein sequences from these genomes. The data in COGs are updated continuously following the sequencing of new prokaryotic genomic sequences.

COGs database is a comprehensive gene-family definition database, developed to classify all conserved genes based on their homologous relationships and evolutionary development [23]. Each COG consists of at least three proteins assumed to have the same evolutionary counterparts. As described by [23], the COGs database is a growing and useful resource to identify genes and groups of orthologs across prokaryotic species that are related by evolution. Information about every completely sequenced and annotated prokaryotic genome is stored in the PTT-formatted files. The PTT file format is a table of protein features, prepared by the National Center for Biotechnology Information (NCBI). The complete collection of current PTT files can be found at ftp://ftp.ncbi.nih.gov/genomes/. Organization of the PTT files is summarized in Table 1 [24].

From every suitable NCBI PTT file, we extracted information about length (column 3) and COG (column 8). We added the genome index and a nominal binary identifier (chromosome, plasmid) to get a file describing the complete set of gene lengths across all available prokaryotic genomes. After the processing of the PTT files, two files were obtained.

One contained the names of genomes with a record format /integer, string/ <genome_index, genome_name> (for example, 22, *Bacillus amyloliquefaciens* dsm 7). The other was a gene-length file with a record format /integer, integer, integer/ <genome_index, COG_index, protein_length> //for example, 2, 1474, 411//. These data were sorted

1	2	3	4	5	6	7	8	9
Location	**Strand**	**Length**	**PID**	**Gene**	**Synonym**	**Code**	***COG***	**Product**
16..1251	+	**411**	284161129	-	Arcpr_0001	-	COG**1474**LO	orc1/cdc6 family replication initiation protein
1251..1961	+	**236**	284161130	-	Arcpr_0002	-	COG**0179**Q	5-carboxymethyl-2-hydroxymuconatedelta-isomerase

Line 1: Description of sequence to which the features belong, e.g. " Archaeoglobusprofundus DSM 5631chromosome, complete genome-1..1560622".
Line 2: Number of proteins, e.g. "1819 proteins"
Line 3: Column headers, tab separated, e.g. "Location Strand Length PID Gene Synonym Code COG Product"
Line 4 onwards: Feature lines, nine columns, tab separated, "-" used for empty fields:

Table 1: Format of PTT files.

by COG_index, genome_index, protein_length in ascending order.All currently available genomes were described in these two files. To check the ranking procedures described below we used small subsets of this dataset.

Pre-processing procedures

To get an input file for further ranking the following pre-processing procedures were applied:

1. Selection of subsets of genomes. A subset may be defined applying different criteria: it may be either a representative sample, a taxa-specific subset, or randomly chosen genomes.

2. Application of a filtering parameter (an entry threshold) on a selected subset. Only COGs containing more than a threshold number of genomes are considered for further processing. For example, if the filtering value is equal to 20% and an amount of genomes in a subset is equal to 500, then only COGs containing at least 100 genomes are considered (passed the entry threshold).

3. Sampling: If there are multiple instances of a COG related to the same genome, a median length value for all paralogs (triplets <genome_index, COG_index, protein_length> from the same genome and from the same COG) is used for further processing.

Set of genomes

To compare performance of the methods, we used the same dataset as in our previous publication [3]. This small set contains 9 *Archaeal* and 91 *Bacterial* genomes. Table 2 of [3] briefly describes these genomes.

Average ranking method (A-rank)

Given a matrix A_{MxN} where $A_{i,j}$ is the value of j^{th} descriptor of the i^{th} object, the average ranking method works this way: for each object *i* the average ofall its descriptor values are calculated, which determines the rank of object *i* relative to other objects. All missing values are ignored.

Simple additive ranking (SAR)

SAR or, alternatively, SAW (Simple Additive Weighting) is the oldest, most widely known and practically used method [25,26]. The method integrates several criteria into a single-weighted value. This is reflected in its name. In the framework of this method the object gets its rank through a simple addition of weighted ranks obtained by sorting of individual attributes. First, weights based on the importance of various attributes are assigned. Second, the ranks within the attribute are scaled. This means that whatever the ranges of the intra-ranks of the particular intra-attribute rankings are, they should each be converted to a comparable scale. For example, if there are 100 objects, one attribute has a ranking scale from 1 to 10 and the other has a ranking

	Filtered Matrix (100×1455)			
Genome Name	**SAR-rank**	**A-rank**	**B-rank**	**SA-rank**
bacillus cereus atcc 14579	1	6	16	17
campylobacter concisus 13826	2	12	6	9
thermotoga sp. rq2	3	26	3	2
aquifexaeolicus vf5	4	15	12	8
campylobacter curvus 525.92	5	17	9	12
bacillus cytotoxicusnvh 391-98	6	5	19	19
dictyoglomus thermophilum h-6-12	7	10	13	14
thermotoganeapolitanadsm 4359	8	19	5	5
bacillus amyloliquefaciensdsm 7	9	13	22	24
helicobacter felisatcc 49179	10	38	10	13
caldicellulosiruptor bescii dsm 6725	11	34	14	23
listeria monocytogenes serotype 4b str.	12	11	15	20
archaeoglobus fulgidusdsm 4304	13	1	1	1
francisella sp. tx077308	14	36	21	28
thermoplasmavolcanium gss1	15	8	2	4
thermoplasmaacidophilumdsm 1728	16	9	4	6
.	.	.	.	.
….	…	…	…	…
.	.	.	.	.
gluconacetobacter diazotrophicus pal 5	88	83	90	88
starkeya novella dsm 506	89	72	87	87
bifidobacteriumanimalis subsp. lactis ad011	90	99	100	100
rhodopseudomonas palustris dx-1	91	76	88	89
rhodospirillumcentenumsw	92	92	89	90
burkholderiarhizoxinicahki 454	93	60	97	96
intrasporangiumcalvumdsm 43043	94	91	95	93
rhodopseudomonas palustris py2	95	90	91	91
rothiadentocariosaatcc 17931	96	97	99	99
streptomycesgriseus subsp. griseusnbrc 3350	97	93	93	95
streptomycesscabiei 87.22	98	96	94	94
salinibacterruber m8	99	98	98	98
haliangium ochraceumdsm 14365	100	100	96	97

Table 2: Results of rankings obtained by the SAR-ranking, Average ranking (A-rank), Bubble sorting (B-rank), and Simulated Annealing (SA-rank). List of genomes in the SA ranking order. Only top- and bottom-ranked genomes are shown.

scale from 1 to 55, they both must be converted to the scale [1:100]. Once all attribute rankings cover the same scale, they can be multiplied by their respective attribute weights. The "utility" for each object is defined by adding the scaled weighted scores across the attributes with further dividing by the number of contributing attributes. Objects are sorted in order of ascending utility.

Here is how SAR strategy is applied to the COGs dataset. All COGs are assigned a number. Given a matrix A_{MxN} where $A_{i,j}$ is the value of j^{th} descriptor of the i^{th} object, the ranking is based on an individual ranking of each object based on the weighted sum of ranks for each descriptor separately. Thus, the eventual rank of the i^{th} object, R_i, is calculated as

$$R_i = \frac{1}{M}\sum_{j=1}^{N} w_j r_{ij}$$

Where j refers to descriptor number and w_j is the weight of the j^{th} descriptor and r_{ij} the ranking of the i^{th} object with regard to the j^{th} descriptor. Subsequently the ranks are scaled (normalized). In case of a sparse matrix, r_{ij} of a missing descriptor value takes the fixed rank value M/2 while rij of non-missing descriptor values are calculated regularly and then scaled uniformly to the range [1..M]. Note that we used the version in which w_j=1 for every j.

Bubble-sort ranking (B-rank)

The strategy of Local Pairwise Interchange (LOPI) amounts to choosing a pair of objects, interchanging them, and evaluating a quality of ranking t(ψ) for the changed ranking [27]. If the new ranking is better than a previous one, then the new ranking is accepted. The procedure is stopped if there is no interchange that may improve t(ψ). LOPI does not guarantee global optimality, but it is very efficient [28], and being enhanced by the Monte Carlo technique[17] brings sufficiently good results. (Randomness is introduced through the random choice of the initial configuration.) In a simulation study by [28] the LOPI strategies found a global maximum of t (ψ) in the majority of the cases.

As a LOPI strategy we apply here the regular "bubble sort" procedure [29] interchanging the rows of a given matrix. The order of any two rows, r_1 and r_2, under this method is determined by the sign of the scalar $G(r_1,r_2)$, such that if $G(r_1,r_2)<0$, then r_1 precedes r_2, and if $G(r_1,r_2)>0$, then r_2 precedes r_1, otherwise no change to their original order.

$$G(r_1,r_2)=\sum_{i=1}^{N}(\eta(r_1^i-r_2^i)-\eta(r_2^i-r_1^i))$$

Where step function $\eta(x)=1$ if $x>0$ and $\eta(x)=0$ otherwise.

Simulated annealing

For large datasets an exhaustive search is computationally demanding and not feasible. For example, in order to use direct maximization for ranking of N=1390 genomes, one needs to examine $N! \approx 10^{103765}$ configurations and calculate "system energy" for every configuration. Because of this, heuristic methods such as the Simulated Annealing method [21] are frequently used. Simulated Annealing Procedure is a general probabilistic meta-heuristic for the global optimization problem of obtaining a good approximation to the global optimum of a given function in a large search space.

Simulated Annealing models a process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy. At each step, a pair of objects is randomly chosen, than the objects are interchanged, and a quality of ranking τ for the changed ranking is evaluated. The algorithm accepts the proposed ranking that increases τ, but also, with a certain probability, rankings that lower the τ. We used acceptance probability function in the form $\alpha(\tau,\tau_{new},t)=\min\left(1,e^{\frac{\tau_{new}-\tau}{t}}\right)$ The algorithm was implemented in R using the mpiR package to enable parallel processing using a high performance computer cluster for large datasets. By accepting points that lower the objective, the algorithm avoids being trapped in local maximum in early iterations and is able to explore globally for better solutions.

Results and Discussion

The Average ranking, the Simple Additive Ranking, and the Bubble sort ranking methods were applied both to the non-filtered input (matrix of size 100×5664, Figure 1) and to a smaller matrix (filtered version: excluding columns that contain more than 65% null values given a matrix of size 100×1455, Figure 2). Simulated Annealing procedure was applied only to the smaller matrix because of computational complexity. Each procedure produced a certain order of rows in matrix, a ranking vector *X*. Calculating of the Kendall tau correlation coefficients between the ranking vector *X* and each column of the matrix yielded distribution of correlations between the global ranking and individual COGs. These distributions are shown in Figures 1 and 2.

Distribution of τ' before the ordering (random ranking) is shown as a blue line and the post-ordering distributions are shown as bar graphs. Unordered genomes produce distributions of τ that are centered at zero. Application of all four ranking algorithms to this set resulted in right-shifted distributions of τ. The distributions of τ in Figures 1 and 2 are bell-shaped and are similar to each other, and the post-ordering distributions were all distinctly shifted to the right.

We conducted the Shapiro-Wilk test of normality for the unordered genomes and found that p-value is 0.19>0.05, supporting our visual examination that the randomly ordered genomes have approximately normal distribution of τ. All ordered distributions fail the normality test (p-value<0.05). For the unordered distribution, the skewness is 2.0703. The ordered distributions have larger absolute values of skewness, which are ≈ -5 (Figure 1).

Since the distributions in Figures 1 and 2 are approximately normal and our sample sizes are large, we used t-test to find the significance of the differences of the means (shown in Table 3). Difference between the means of ordered and unordered sets is around 0.2 for all four methods (p-value $<10^{-16}$ for all four comparisons) [6,30]. It should also be mentioned that (a) all τ values for the filtered matrices are higher than their counterparts calculated for the complete matrices, and (b) the Simulated Annealing approach results in the best ordering, with Bubble sort being a close second [31-34] (Table 3).

It has already been mentioned [3,4,35] that A-rank is not likely to produce valuable results as compared with other ranking methods mentioned in this study. The difference between Simulated Annealing and Bubble Sort ranking is 0.002, which is not significant, given p-value of the t-test is 0.9854. We do not expect different ranking methods to yield identical genome orders, since different methods use different criteria for ordering.

Significant differences in genome orders produced by different ranking methods are expected, since the four methods use different criteria for ranking. However, we find that the four methods are highly correlated (Table 4). B-rank (Bubble Sort) and SA-rank (Simulated

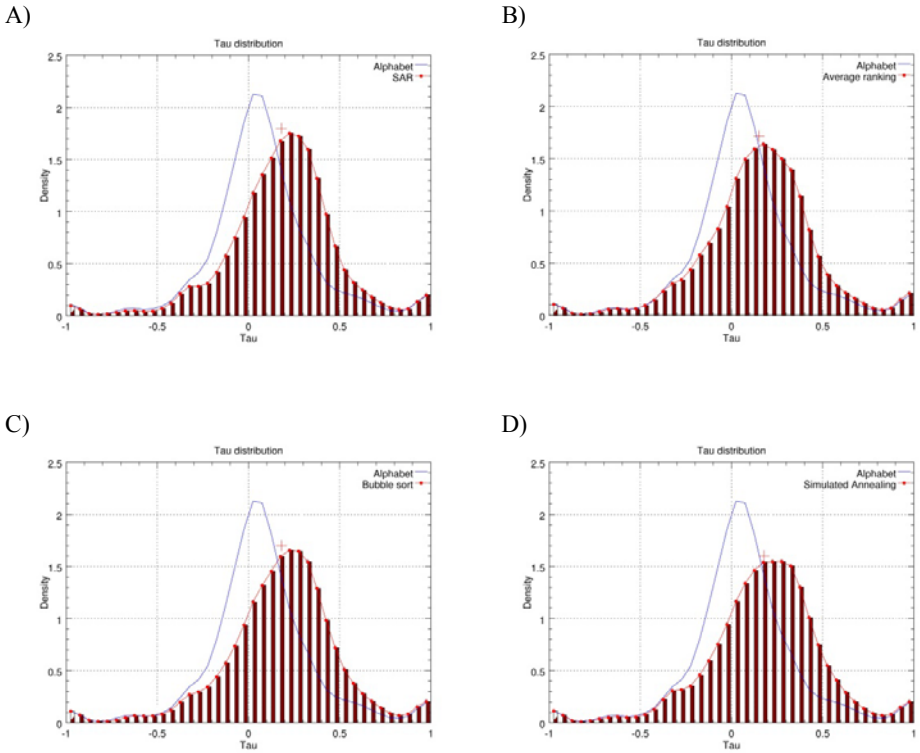

Figure 1: Histograms of the Kendall tau correlation coefficients between the ranking vectors and vectors of gene lengths of individual COGs obtained using the input matrix of size 100×5664. The blue lines show histograms of the Kendall tau correlation coefficients between a random ranking (using the alphabetic order of the 100 bacteria names as a random control) and individual COGs. In red the histograms of τ's obtained by various ranking methods are shown: A) Simple Additive ranking B) Average ranking C) Bubble Sort, and D) Simulated Annealing. The red + shows the average value of Tau in the various methods.

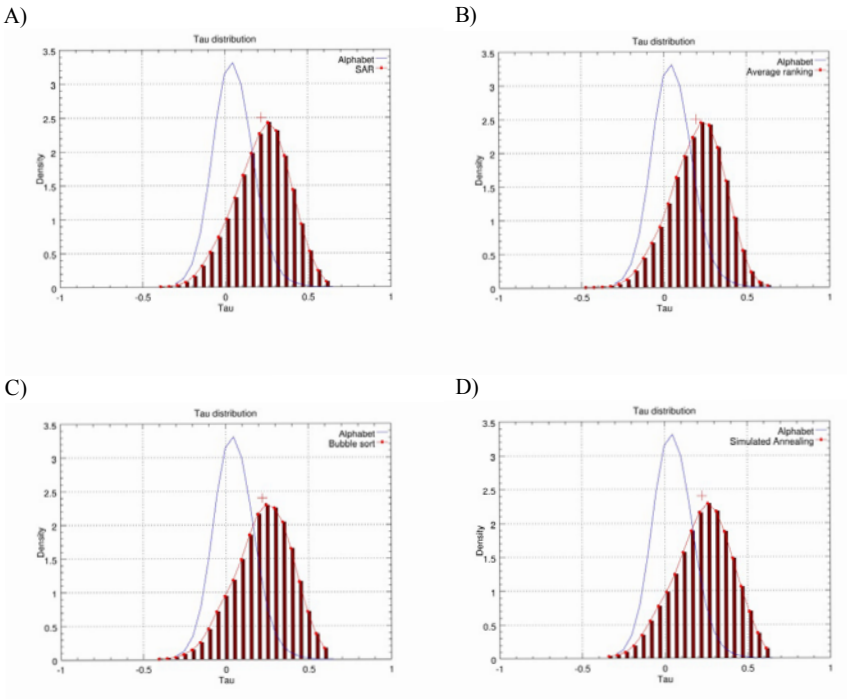

Figure 2: Histograms of the Kendall tau correlation coefficients between the ranking vectors and vectors of gene lengths of individual COGs obtained using the input matrix of size 100×1455 (Filtered version: excluding columns that contain more than 65% null values). Usage of colors, line and bar graphs is as in Figure 1. The red + shows the average value of the distributions obtained by the various methods.

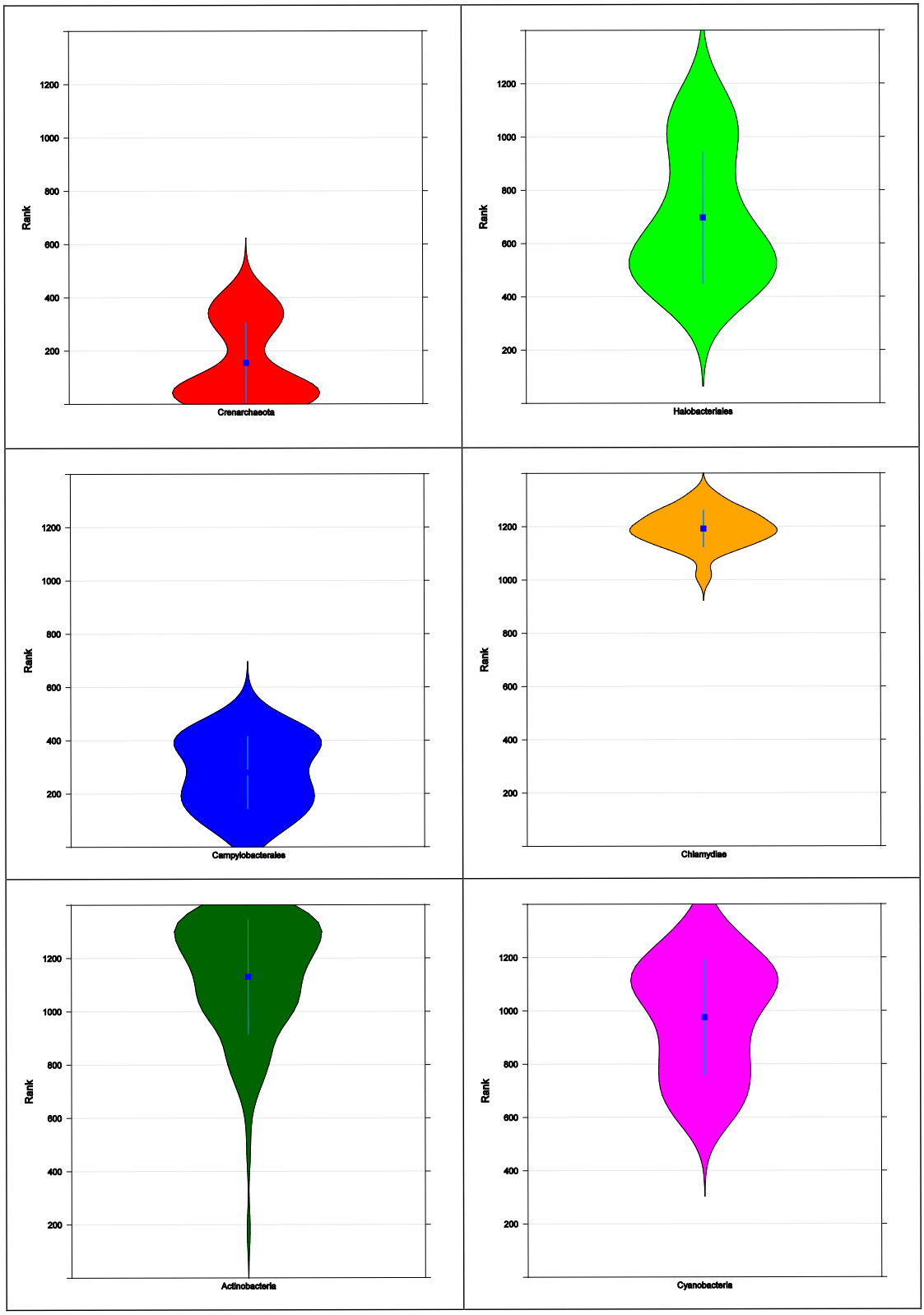

Figure 3: Violin plots of length distributions for six groups of prokaryotic genomes. Ranks calculated by applying Bubble sort to the filtered set of 1390 prokaryotic genomes.
Top: Crenarchaeota (left), Halobacteriales (right)
Middle: Campylobacterales (left), Chlamydiae (right)
Bottom: Actinobacteria (left), Cyanobacteria (right)

	Tau sortedness	
Ranking Method	**Complete Matrix (100× 5664)**	**Filtered Matrix (100×1455)**
Average Ranking	0.15114	0.19459
Simple Additive Ranking	0.17990	0.21736
LOPI (Bubble Sorting)	0.18048	0.22057
Simulated Annealing		
(applied to the filtered matrix)	0.17841	0.22381

Table 3: Goodness of fit of rankings measured by Kemeny measure.

	SAR-rank	A-rank	B-rank	SAR-rank	A-rank	B-rank	SA-rank
SAR-rank	1						
A-rank	0.47556	1					
B-rank	0.74949	0.47152	1				
SAR-rank (filtered)	0.87475	0.52081	0.80121	1			
A-rank (filtered)	0.5899	0.77253	0.56566	0.64242	1		
B-rank (filtered)	0.7503	0.48848	0.95313	0.81333	0.58424	1	
SA-rank (filtered)	0.73293	0.53333	0.81212	0.80646	0.63313	0.84444	1

Table 4: Pairwise Kendall coefficients of correlation between different rankings.

Annealing) are the most similar rankings (τ=0.84 for the filtered datasets). Table 4 also shows that filtering improves the agreement between different ranking methods by removing less abundant COGs that skew the results.

For all four methods there is significant average correlation between global rankings and individual COGs lengths. The filtering procedure results in the distributions without heavy left and right tails. COGs in the two extremes of the distribution have a small number of genomes and addition of these scores to the total τ may create a bias. The comparison of Figures 1 and 2 clearly demonstrates necessity of a filtering stage among the preprocessing procedures. We should not be surprised by this observation, since have already demonstrated that COGs containing very small amount of genomes ("unpopulated" COGs) should not be considered in phylogenomic methods. The "Forest of Life" concept was developed by [31-34]. They showed that there is a general evolutionary trend in the "Forest of Life" as well as in the "Tree of Life", but gene trees constructed from unpopulated COGs poorly contribute to and frequently contradict it. Now we demonstrated that the same is true while ranking prokaryotic genomes-unpopulated COGs should be excluded. Note that although histograms A-D in Figure 2 are very similar, this does not imply identical genome orderings.

Table 2 shows the genomes in ascending order as dictated by the SAR technique: genomes with shorter genes occupy the top portion of the table. Several features of the Table 2 are quite remarkable. First, the columns are not identical; moreover, they are not even nearly identical. Nevertheless, the SAR ranking at the bottom part (longer genes) amazingly coincides with all of the other three ranking methods. This is not the case for the top part of the SAR ranking. While all hyperthermophiles that have high SAR ranks (*Thermotogae, Thermoplasmata, Aquifex,* and *Archaeoglobus*) also have high SA and B ranks as well, Bacilli that have high SAR ranks (ranks 1, 6, 9) have much lower ranks by the SA ranking method (17, 19, 24, correspondingly).

Performance for Bubble Sort is $O(n^2)$, and for Simulated Annealing is $O(K(n^2+n) \log n)$ [35,36], where K is the number of COGs and n is the number of genomes. Therefore, we conclude that Bubble Sort is faster than Simulated Annealing and produces nearly identical ranking (Table 4).

As we can see from the Table 3, the Average ranking method results in the lowest value of sortedness (0.15114 for the complete COG's dataset and 0.19459 for the filtered dataset). It also has the lowest correlation, in the vicinity of 0.5, with the other three methods (Table 4).

Motivated by the performance of Bubble sort for the 100-genome dataset, we applied it to the set of 1390 genomes. We used the same 35% filtering cut-off as for the 100-genome dataset. Here we show some preliminary results of ordering the big set of genomes (Figure 3) for several groups of genomes.

- *Crenarchaeota*, phylum of *Archaea*, tend to have shorter genes (Figure 3, top left). Almost all sequenced genomes of this phylum are related to hyperthermophilic species. Probably, it is the main factor affecting gene lengths of the species of this phylum, but contribution of other factors is an open question.
- *Halobacteriales* have longer genes (Figure 3,top right).In taxonomy, the *Halobacteria* are a class of the *Euryarchaeota*, and the extremely halophilic, aerobic members of *Archaea* are classified within the family *Halobacteriaceae*, order *Halobacteriales*. So, at the top of Figure 3 we present plots describing distribution of ranks of genomes from two different groups of Archaea. Our speculation is that hyperthermophilicity is a factor of gene shortening, while halophilicity is factor acting in the opposite direction.
- Middle plots of Figure 3 present *Campylobacterales* (left) and *Chlamydiae* (right). All genomes of these two selections are pretty short; however, the plots are very different. Family of *Campylobacterales* (belonging to the phylum *Proteobacteria*), have an average rank of 203, with the smallest rank of 10 (*Helico bacterbizzozeronii* ciii-1) and the largest rank of 392 (*Helicobacter hepaticus ATCC* 51449). Members of the class of *Chlamydiae* have exceptionally long genes: the ranks of 21 members of this class are located from positions 835 to 1127 in the ranking list.
- Bottom plots of Figure 3 present *Actinobacteria* (left) and *Cyanobacteria* (right). As a rule, members of the class of *Actinobacteria* have long genes. Finally, Cyanobacteria also tend to have long genes. At this point, it is only observation and we have no speculations on this issue.

Conclusions

We have presented four methods of genome ranking and compared their performance using a dataset of 100 genomes randomly selected from the entire NCBI collection of *Eubacterial* and *Archaeal* genomes. We have demonstrated that all four methods produce consistent results and that Bubble Sort and Simulated Annealing have the best ranking. Given computational advantages of Bubble Sort, it is the optimal method for the task of genome ordering. We also showed that filtering procedure (removal of the less populated COGs) improve the final sortedness of the dataset.

Using the subset of 100 randomly selected genomes, we demonstrated that hyperthermophilic species have shorter genes than the mesophilic species. Addition of all currently sequenced genomes

did not change this conclusion. It would be wrong to claim that environmental stress always causes genes to be shorter.

Acknowledgements

TT was supported by NIH: GM068968 and NIH-NICHD: HD070996.

References

1. Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, et al. (2003) The COG database: an updated version includes eukaryotes. BMC Bioinformatics 4: 41
2. Tatusov RL, Galperin MY, Natale DA, Koonin EV (2000) The COG database: a tool for genome-scale analysis of protein functions and evolution. Nucleic Acids Research 28:33-36.
3. Bolshoy A, Tatarinova T (2012) Methods of combinatorial optimization to reveal factors affecting gene length. Bioinformatics and Biology Insights 6: 317-327.
4. Hochbaum DS, Moreno-Centeno E, Yelland P, Catena RA (2011) Rating customers according to their promptness to adopt new products. Operations Research 59: 1171-1183
5. Bolshoy A, Volkovich Z, Kirzhner V, Barzily Z (2010) Genome Clustering: from Linguistic Models to Classification of Genetic Texts. Berlin Heidelberg: Springer-Verlag.
6. Bolshoy A, Volkovich Z (2009) Whole-genome prokaryotic clustering based on gene lengths. Discrete Applied Mathematics 157: 2370-2377.
7. Korenblat K, Volkovich Z, Bolshoy A (2012) Robustness of the whole-genome prokaryotic clustering based on gene lengths. Computational Biology and Chemistry 40: 20-29.
8. Kemeny JG (1959) Mathematics without numbers. Daedalus 88: 571-591.
9. Kemeny JG, Snell J (1962) Mathematical Models in the Social Sciences.
10. Ergun F, Jowhari H (2008) On distance to monotonicity and longest increasing subsequence of a data stream. In nineteenth annual ACM-SIAM symposium on Discrete Algorithms. pp730-736.
11. Condorcet MJ (1785) Éssaisurl'application de l'analyse à la probabilité des décisionsrendues à la pluralité des voix.
12. Borda JC (1781)Mémoiresur les èlections au scrutin. Histoire de l'Académie Royale des Sciences.
13. DworkC, Kumar R, Naor M, Sivakumar D (2001) Rank aggregation methods for the web. Proceedings of the 10th international conference on World Wide Web: 613-622.
14. Fagin R, Kumar R, Sivakumar. D (2003) Efficient similarity search and classification via rank aggregation. SIGMOD 301-312.
15. BartholdiJ III, Tovey CA, Trick MA (1989) Voting schemes for which it can be difficult to tell who won the election. Social Choice and Welfare 6:157-165.
16. Diaconis P, Graham R (1977) Spearman's footrule as a measure of disarray. Journal of the Royal Statistical Society Series B 39: 262-268.
17. Metropolis N, Ulam S (1949) The Monte Carlo method. J Am Stat Assoc 44: 335-341.
18. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of State Calculations by Fast Computing Machines. Journal of Chemical Physics 21: 1087-1091.
19. Metropolis N (1987) The beginning of the Monte Carlo method. Los Alamos Science Special Issue 125-130.
20. Hastings W (1970) Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika 57: 97-109.
21. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 220: 671-680.
22. Cerný V (1985) Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm Journal of Optimization Theory and Applications 45: 41-51.
23. Tatusov RL, Koonin EV, Lipman DJ (1997) A genomic perspective on protein families. Science 278: 631-637.
24. Perldoc (pdoc rendered) documentation for bioperl modules. Bio Perl.
25. Tzeng GH, Huang JJ (1981) Multiple Attribute Decision Making: Methods and Applications. CRC Press, USA.
26. Kropp J, Scheffran J (2007) Advanced Methods for Decision Making and Risk Management in Sustainability Science. Nova Science Publishers, USA.
27. Borg I, Groenen PJF (2005) Modern Multidimensional Scaling: Theory and Applications. Springer Publishers, USA.
28. Groenen P (1993) The majorization approach to multidimensional scaling: Some problems and extensions. DSWO Press, Netherlands.
29. Knuth DE (1973) Art of Computer Programming. Addison-Wesley, USA.
30. Korenblat K, Volkovich Z, Bolshoy A (2012) Robust classifying of prokaryotic genomes. Comput Biol Chem 40: 20-29.
31. Koonin EV, Puigbò P, Wolf YI (2011) Comparison of phylogenetic trees and search for a central trend in the "forest of life". J Comput Biol 18: 917-924.
32. Puigbò P, Wolf YI, Koonin EV (2009) Search for a 'Tree of Life' in the thicket of the phylogenetic forest. J Biol 8: 59.
33. Puigbò P, Wolf YI, Koonin EV (2010) The tree and net components of prokaryote evolution. Genome Biol Evol 2: 745-756.
34. Puigbò P, Wolf YI, Koonin EV (2013) Seeing the Tree of Life behind the phylogenetic forest. BMC Biol 11: 46.
35. Hochbaum DS, Levin A (2006) Methodologies and algorithms for group-rankings decision. Management Science 52: 1394-1408.
36. Hansen PB (1992) Simulated annealing. Electrical Engineering and Computer Science Technical Reports. Paper 170.