

RESEARCH

Open Access

On the number of genomic pacemakers: a geometric approach

Sagi Snir

Abstract

The universal pacemaker (UPM) model extends the classical molecular clock (MC) model, by allowing each gene, in addition to its individual intrinsic rate as in the MC, to accelerate or decelerate according to the universal pacemaker. Under UPM, the relative evolutionary rates of all genes remain nearly constant whereas the absolute rates can change arbitrarily. It was shown on several taxa groups spanning the entire tree of life that the UPM model describes the evolutionary process better than the MC model. In this work we provide a natural generalization to the UPM model that we denote multiple pacemakers (MPM). Under the MPM model every gene is still affected by a single pacemaker, however the number of pacemakers is not confined to one. Such a model induces a partition over the gene set where all the genes in one part are affected by the same pacemaker and task is to identify the pacemaker partition, or in other words, finding for each gene its associated pacemaker. We devise a novel heuristic procedure, relying on statistical and geometrical tools, to solve the problem and demonstrate by simulation that this approach can cope satisfactorily with considerable noise and realistic problem sizes. We applied this procedure to a set of over 2000 genes in 100 prokaryotes and demonstrated the significant existence of two pacemakers.

Keywords: Molecular evolution, Genome evolution pacemaker, Deming regression, Partition distance, Gap statistics

Background

The Universal PaceMaker (UPM) of genome evolution [1] extends the classical Molecular Clock (MC) model [2] and its various imperative relaxations (see e.g. [3,4] among a few), by relaxing the rate constancy (as in MC) on one hand, and yet preserving the rate correlation between the various genes. Such a model can provide explanation to the striking phenomenon that the distribution of the evolutionary distances between orthologous genes remains remarkably constant across the entire history of life [5-7]. Under the UPM, all genes in each evolutionary lineage adhere to the pace of a pacemaker (PM), and change their evolutionary rate (approximately) in unison although the pacemaker's pace at different lineages may differ. The UPM model is compatible with the large amount of data on fast-evolving and slow-evolving organismal lineages, primarily different groups of mammals [8]. Alternatively, the constancy of gene-specific relative rates is also an outcome of the MC model, under which the different genes evolve at roughly constant albeit different (gene-specific)

rates. In a line of works [1,9,10] we established the superiority of the UPM model over the MC by explaining a larger fraction of the variance in the branch lengths of thousands of gene trees spanning the entire tree of life. Despite its relative simplicity, it was noted [11] that the UPM is “the most plausible model of genomic evolution and appears to have some statistical support”.

Although highly statistically significant, in absolute terms however, the advantage of UPM over MC was small, and both models exhibited considerable evolution rate overdispersion. A plausible explanation to the latter is that instead of a single, apparently weak (overdispersed) PM, there are independent *multiple pacemakers* that each affect a (different) subset of genes and are less dispersed than the single pacemaker. Throughout, we use the notation UPM to refer to the model and the PM term for the pacemaker as an object.

Primarily, we investigate the requirements for the identification of distinct PMs and assignment of each gene to the appropriate PM. Such an assignment forms a partition over the set of genes and hence we denote this task as the *PM partition identification* (PMPI) problem. PM identification depends on the number of analyzed genes,

Correspondence: ssagi@research.haifa.ac.il
Department of Evolutionary Biology, University of Haifa, Haifa 31905, Israel

the number of target PMs, the intrinsic variability of the evolutionary rate for each gene and the intrinsic variability of each PM. The PMPI problem is theoretically and practically hard as it concerns dealing with a lot of data obscured by a massive amount of noise. A possible direction to pursue is to exploit the signal from the data themselves in order to reduce the search space and focus only on relevant partitions. We note that partitioning over the gene set or even different positions in a gene, has been done before [12–14]. However, to the best of our knowledge, these were not based on a rigorous model as the pacemaker rather mainly on the level of the mutation rates.

In this work, a first attempt in this direction is made by devising and employing a novel technique using a series of analytic tools to solve the PMPI problem, and assess the quality of the derived solution. We tackle theoretical computational and statistical issues, as well as challenging engineering obstacles that arise along the way. These include guaranteeing *homoschedasticity* [15] by working in the log space, removing gene order dependency [16] by employing the Deming regression [17,18], and graph completion through most reliable paths. The result is the partial *gene correlation graph* where edge lengths represent (inversely) correlation, that we subsequently embed into the Euclidean space while preserving the distances (see Figure 1). We apply standard clustering tools to this data and assess the significance of the result. We next formulate the PMPI problem as a recoloring problem [19,20] where a gene's PM is perceived as its color and the (set of) genes associated with a certain PM form a color class. To measure the quality of partition reconstruction, one may look for the minimum number of genes that need to be recolored in order that every part in the reconstructed partition is monochromatic. This number (the recolored genes) is denoted the *partition distance* [21] and can be

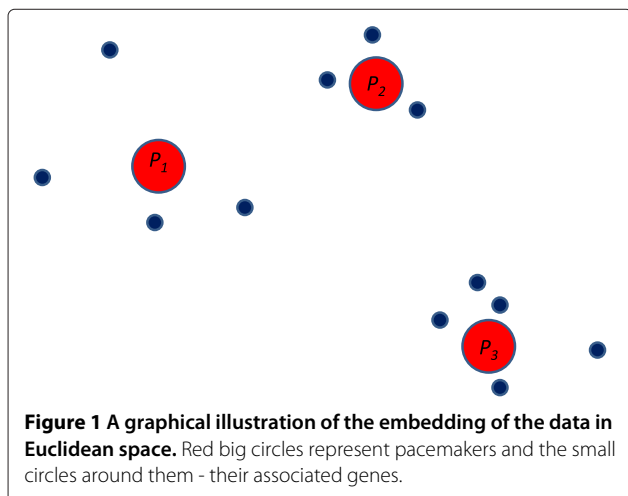
solved by a matching algorithm. We however use a greedy maximum weighted matching algorithm, that is practically simpler for implementation and provided very good results empirically. Although theoretically this algorithm provides a 1/2-approximation guarantee for any input, under some statistical conditions the we note, with high probability the correct partition distance is returned.

The simulation results obtained using this approach are highly significant under a random model that we devise. The latter is significant as it implies that we were successful in both extracting the signal from the (noisy) data, and our technique is plausible. Finally, using insights from the simulation analysis, we analyzed the large set of phylogenetic trees of prokaryotic genes that was previously studied in [1]. Because the actual PM partition is unknown, we used the gap statistics criterion of Tibshirani et al. [22] to determine clustering significance and resulting in identification of two distinct genome evolution PMs.

We end this part with a visual illustration given by a graphical representation in Figure 1 where the big red balls represent PMs and the small circles represent genes. By the illustration, it is clear that the more genes and PMs the more likely they are to mix between each other. Similarly, the larger the distance between a gene and its PM, the greater the probability to misclassify it. Conversely with respect to the distance between PMs - the greater this distance, the less likely gene to be misclassified.

The evolutionary model

Evolutionary history is described by a tree $T = (V, E)$ which is a combinatorial object composed of nodes representing (extant and extinct) species, and edges connecting these nodes such that there are no cycles in T . The edges are directed from an *ancestor* to its *descendant* nodes and also correspond to the time period between the respective nodes. There is one node with no ingoing edges, the *root*, and nodes with no outgoing edges are the *leaves* that are labeled by the *species* (or *taxa*) set. Therefore, the topology of T indicates the history of speciation events that led to the extant species at the leaves of T . Internal nodes correspond to ancestral forms existed at speciation events, and edges indicate ancestral relationships. A node (or a species) is a set of genes $G = \{g_i\}$ where a gene is a sequence of *nucleotides* of some given length. A gene evolves through a process in which mutations change its nucleotides from one state to another. In our model, all extant and extinct species possess the same set of genes $G = \{g_i\}$ and all genes g_i evolve along T according to an evolutionary model that is assumed to follow a continuous time Markov process. This process is represented by a given rate of mutations r per unit of time. In particular, every gene g_i evolves at an intrinsic rate $r_i \in \mathbf{r}$ that is constant along time but deviates randomly along the time



periods (i.e. tree edges). Let $r_{i,j}$ be the *actual* (or *observed*) rate of gene i at period j . Then $r_{i,j} = r_i e^{\alpha_{i,j}}$ where $0 < e^{\alpha_{i,j}}$ is a multiplicative *error factor*. The number of mutations in gene g_i along period t_j is hence $\ell_{i,j} = r_{i,j} t_j$, commonly denoted as the *branch length* of gene g_i at period j . Throughout, we will use i to identify genes g_i and j for time periods t_j . As the topology of T is constant and assumed to be known, we will not make any reference to the tree and regard the edges only as independent time periods t_j for $1 \leq j \leq \tau$ where $\tau = |E|$.

We now extend this model to include a pacemaker that accelerates or decelerates a gene g_i , relative to its intrinsic rate r_i . Formally, a *pacemaker* PM_k is a set of τ *paces* $\beta_{k,j}$, $1 \leq j \leq \tau$ where $\beta_{k,j}$ is the relative pace of PM k during time period t_j and $-\infty < \beta < \infty$. Under the UPM model, a gene g_i that is *associated* with PM P_k has actual rate at time t_j : $r_{i,j} = r_i e^{\alpha_{i,j}} e^{\beta_{k,j}}$. Hence, for $\beta < 0$ the PM slows down its associated genes, for $\beta > 0$ genes are accelerated by their PM , and for $\beta = 0$, the PM is neutral. Assume every gene is associated with some PM and let $PM(g_i)$ be the PM of gene g_i . Then the latter defines a partition over the set of genes G , where genes g_i and $g_{i'}$ are in the same part if $PM(g_i) = PM(g_{i'})$.

Comment 1. *It is important to note that gene rates, as well as pace makers paces, are hidden and that we only see for each gene g_i , its set of edge lengths $\ell_{i,j}$.*

Comment 2. *The presence of two genes in the same part (PM) does not imply anything about their magnitude of rates, rather on their unison of rate divergence.*

The above gives rise to the *PM Partition identification Problem*:

Problem 1. Pacemaker Partition Identification. Given a set of n genes g_i , each with τ branch lengths $\{\ell_{i,j}\}$, the *Pacemaker Partition Identification* (PMPI) problem is to find for each gene g_i , its pace maker $PM(g_i)$.

We first observe the following:

Observation 1. *Assume gene g_i has error factor $\alpha_{i,j} = 0$ for all time periods t_j , $1 \leq j \leq \tau$ and let $P' = PM(g_{i'})$ be the pace maker of gene $g_{i'}$ with relative paces e^{β_j} . Then at all periods t_j , $r_{i,j} = r_i e^{\beta_j}$.*

Observation 1 implies that if genes g_i and $g_{i'}$ belong to the same pace maker, and both genes have zero error factor at all periods, then at all periods, the ratio between the edge lengths at each period is constant and equals to $r_i/r_{i'}$. This however is not necessarily true if one of the error factor is not zero or genes g_i and $g_{i'}$ do not belong to the same

pace maker. Recall that we do not see the gene intrinsic rates (and hence also the ratio between them). However if we see the same ratio between edge lengths across all time periods, we can conclude about the error factors and possibly their belonging to the same PM .

In order to tackle the PM identification problem, we impose some statistical structure (as observed in real data [6]) on the given setting. The goal is to assume that the error factor of each gene is small enough at every period, so that all genes belonging to the same PM , change their actual rate in unison.

Similarly, we assume that β_k varies so that genes from different PM s (parts) can be distinguished (otherwise, no difference except their random error factor exists).

Assumption 1.

1. For all genes g_i and periods t_j , the gene error factors $\alpha_{i,j}$ follow a normal distribution $\alpha_{i,j} \sim N(0, \sigma_G^2)$,
2. For all PM s P_k and periods t_j , the PM paces $\beta_{k,j}$ follow a normal distribution $\beta_{k,j} \sim N(0, \sigma_P^2)$,

The pacemaker partition identification procedure

Here we devise a procedure to solve the PMPI problem that entails a technique to infer distances between genes, constructing the *gene correlation graph*, embed reliably this graph in the plain and apply partitioning algorithms to this embedding. We now describe each of these steps.

Inferring gene distance

As outlined above, our first task is to infer gene pairwise distances from the raw data, which is gene edge lengths $\ell_{i,j}$ for every time period (edge) j . In particular, as the relevant information is encompassed in the random component of that value, the task of extracting that component is even more challenging.

We now proceed as follows: Given two sets of edge lengths $\ell_{i,j}$ and $\ell_{i',j}$ corresponding to genes g_i and $g_{i'}$, and time periods t_j for $1 \leq j \leq \tau$, we draw τ points on a plain $(\ell_{i,j}, \ell_{i',j})$. Now, if the error factors, $\alpha_{i,j} = \alpha_{i',j} = 0$ for all $1 \leq j \leq \tau$ and we connected all these points, we would obtain a straight line. The slope of that line is the multiplicative factor representing the ratio between the rates of evolution of the corresponding genes - $r_{g_i}/r_{g_{i'}}$; we denote it $\rho_{i,i'}$. Obviously, the above description refers to an idealized case. With real data, we never expect to find such a perfect correlation because the characteristic variance σ_G^2 is always non zero. Thus, we expect to find the points scattered around a trend line representing the rate ratio. The density of points around the trend line represents the level of correlation. Our goal is to obtain both the rate ratio $\rho_{i,i'}$ and the level of correlation where the latter will be used to classify between the genes. The method of choice to

pursue here is to apply linear regression [15] between the points representing the two edge lengths. There are several outstanding issues that need to be addressed in such a task.

1. **Zero intercept requirement:** Linear regression, when applied to a set of points on a plane, finds a line $y = ax + b$ minimizing the sum of square distances of that line to all the points. As we deal with a multiplicative factor, the trend line has to cross the origin, i.e. $b = 0$. Hence we need to modify the standard procedure for regression.
2. **Homoscedasticity requirement:** *homoscedasticity* is the property that the error in the dependent variable (y) is identically and independently distributed (IID) along the trend line. However, by our formulation $\ell_{i,j} = t_j r_{i,j} = t_j r_i e^{\alpha_{i,j}}$ and the expected value (the value on the trend line) is $t_j r_i$. The deviation then is $t_j r_i (e^{\alpha_{i,j}} - 1)$. As r_i is constant for all time period, we see that the longer the time period t_j , the larger the influence of $\alpha_{i,j}$. That is, assume two time periods j and j' with the same error factor $\alpha_{i,j} = \alpha_{i,j'}$ but different period lengths, WLOG $\ell_j < \ell_{j'}$. We obtain different deviations $t_j r_i (e^{\alpha_{i,j}} - 1) < t_{j'} r_i (e^{\alpha_{i,j'}} - 1)$, creating a bias toward longer periods. The following observation follows immediately from the definition of $\alpha_{i,j}$.

Observation 2. *If we take the log $\ell_{i',j} = \log t_j r_{i'} + \alpha_{i',j}$ we arrive at Homoscedasticity.*

We denote this as the *log transformation* and also observe the following:

Observation 3. *Under the log transformation the trend line $\log \ell_{i',j} = a \log \ell_{i,j} + b$ has slope one ($a = 1$) and intercept $b = \log \rho_{i,i'}$.*

We will use these properties in our calculations.

3. **Gene order independence:** The final problem with the linear regression has to do with the basic assumptions in least squares analysis. In standard least squares, the assumption is that the independent variable x is error-free while only the dependent variable y deviates from its expected values. In our case, however, the choice between the variables is arbitrary and both are subjected to deviation, according to their characteristic variance σ_G^2 . Handling this case with standard least squares would cause arbitrary bias due to the selection of the variables [16]. To handle this case, we apply Deming Regression [17,18]. This approach assumes an explicit probabilistic model for the variables and extracts closed forms expressions (in the observed

variables) for the sought expected values. To adjust to our specific case, we will use the observations drawn above. The linear model assumed is of type $\eta = \alpha \xi + \beta$ where the observations of both η and ξ , (x_1, \dots, x_n) and (y_1, \dots, y_n) , respectively, have normally distributed errors: (i) $x_i = \xi_i + \varepsilon_{x_i}$, and (ii) $y_i = \eta_i + \varepsilon_{y_i} = \alpha + \beta \xi_i + \varepsilon_{y_i}$. As can be seen, this is exactly our case. The likelihood function of this model is:

$$f = \prod_1^n (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{(x_i - \xi_i)^2}{2\sigma^2}\right) \times (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{(y_i - \alpha - \beta\xi_i)^2}{2\sigma^2}\right) \quad (1)$$

Under the general formulation, the ML value for α is: $\alpha = \bar{x} + \bar{y}\beta$ where \bar{x} and \bar{y} are the average values for x_i and y_i . However, in our formulation we have $\beta = 1$ and hence $\alpha = \bar{x} + \bar{y}$. Having α at hand, we can reconstruct the trend line and obtain the deviation of every point from it. Finally, by our formulation, $\rho_{i,i'}$ is given by $\exp(\alpha)$ and the correlation between the rates is the standard *sample Pearson correlation coefficient* $r(X, Y)$ [15]:

$$r = \frac{\sum_{i=1} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1} (X_i - \bar{X})^2} \sqrt{\sum_{i=1} (Y_i - \bar{Y})^2}} \quad (2)$$

The gene correlation graph

After we inferred all pair-wise correlations, we can build the *Gene Correlation Graph* $G = (V, E, w)$ aiming at representing the correlation between the pairs of genes. $V = \{g_i\}$ and an edge $(i, i') \in E$ if $r(i, i')$ from Equation (2) is greater than some threshold δ_r , maintaining a minimal level of correlation in the graph. Hence we set $w(i, i') = r(i, i')$ and as $-1 \leq r \leq 1$ we are guaranteed no negative weighted edges exist. Note that we are not interested in r^2 which may reflect high *negative* correlation, rather only in high positive correlation.

Recall that our initial goal was to partition genes into clusters (PMs) according to correlation. Perhaps the most commonly used technique is *k-means* [23,24] that aims at minimizing the within-cluster sum of squares (WCSS). These techniques operate in the Euclidean space and hence some distance preserving technique is required to embed the correlation graph G in the space. Multi-dimensional Scaling [25] (also Euclidean embedding) is a family of approaches for this task. Kruskal's iterative algorithm [26] for non-metric multidimensional scaling (MDS) receives as input a (possibly partial) set of distances and the desired embedding should preserve the *order* of the original distances. It requires however a full matrix as a starting guess.

Our approach here is to join every two nodes by the most reliable connection and with the highest correlation. This translates to finding the path with the minimum number of nodes (hops) and that the multiplication of the corresponding weights is minimal. This distance measure, min hop min weight (MHMW), is also useful in communication networks, where hop distance corresponds to reliability [27].

While the naive algorithm for the latter runs in time $O(n^3)$ it can be easily seen that we can solve the problem in time $O(n^2 \log \text{diam}(G))$ where $\text{diam}(G)$ is the diameter of G . The completed graph \hat{G} serves as input to the *Classical multidimensional scaling* (CMDS) [28] whose output serves as the initial guess to the Kruskal's non-metric MDS. Once we have the embedding, we can apply k -means and obtain the desired clustering.

Below is the complete formal procedure *PMPI*:

Procedure *PMPI*(\mathcal{G}, δ_r):

1. Set the correlation graph $G = (V, E)$ with $V = \emptyset$, $E = \emptyset$
2. $V = \{g | g \text{ is a gene in } \mathcal{G}\}$
3. for all $g_i, g_j \in \mathcal{G}$
 - apply the Deming regression between g_i and g_j to determine $r(g_i, g_j)$
 - if $r(g_i, g_j) \geq \delta_r$, then add $\{(g_i, g_j)\}$ to E and set $w(g_i, g_j) \leftarrow r(g_i, g_j)$
4. $\hat{G} \leftarrow \text{MHMW}(G)$
5. apply Classical Multidimensional Scaling (cmdscale) to the full graph \hat{G}
6. apply Kruskal's iterative algorithm (isoMDS) to the original distance matrix, starting from cmdscale output
7. apply k means to the resulted embedding

Simulation analysis

In order to evaluate the *PMPI* procedure described in Section 1 and derive practical intuition over our model, we performed simulation according to the basic lines described above.

In a simulation study, a crucial part involves the assessment of the reconstruction quality with respect to the model on which the input was generated. As the *PMPI* is targeted at reconstruction of the original PM partition, we chose to use the partition distance measure.

Partition distance

Once we obtain the reconstructed clustering, it should be compared to the original, model clustering. The task of comparing two clusterings can be casted as a partition distance where every clustering is a partition over the element set. We now define it formally. For two sets s_i and

s_j , the distance $d(s_i, s_j)$ is the size of their symmetric difference set $s_i \Delta s_j = (s_i \setminus s_j) \cup (s_j \setminus s_i)$. Analogously, the *similarity* $s(s_i, s_j)$ is the size of their intersection set $s_i \cap s_j$ and it is easy to see that given the sizes of the two sets, one is derived from the other. A *partition* \mathcal{P} over a ground element set N is a set of parts $\{p_i\}$ where every part is a subset of N , $\{p_i\}$ are pairwise disjoint (i.e. $p_i \cap p_j = \emptyset$ for every $i \neq j$), and their union is N . The cardinality of \mathcal{P} , denoted as $|\mathcal{P}|$ is the number of parts. A partition can also be perceived as a *coloring* function C from N to a set of colors \mathcal{C} (the *color classes*) where $C(x)$ is the part of element $x \in N$ under partition \mathcal{P} (or equivalently C). Henceforth we will use the notions of PM identity and a color interchangeably. Given two partitions \mathcal{P} and \mathcal{P}' over the same element set N (or equivalently C and C'), denoted as the *source* and *target* partitions, we are interested in their *partition distance* $d(\mathcal{P}, \mathcal{P}')$ as some measure of similarity. The simplest approach is naturally the number of elements with different colors at the two partitions, i.e., $x \in N$ s.t. $C(x) \neq C'(x)$, and we call it the *identity similarity*. Under this approach, the partition distance between \mathcal{P} and \mathcal{P}' , $d(\mathcal{P}, \mathcal{P}')$, is defined as:

$$d(\mathcal{P}, \mathcal{P}') = \sum_{x \in N} \bar{\delta}(C(x), C'(x)) \quad (3)$$

where $\bar{\delta}$ is the *inverse Kronecker delta*:

$$\bar{\delta}(i, j) = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

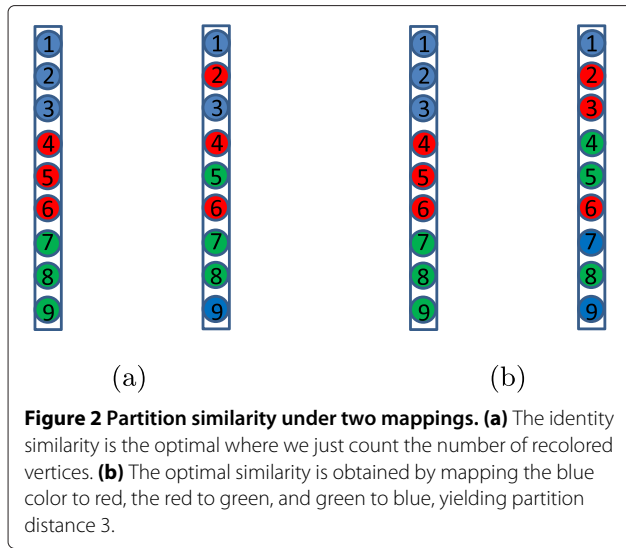
This of course is simple and is an upper bound on a more accurate approach: colors can be permuted between the two partitions, in the sense that a color is mapped by a function f to another color in \mathcal{C} and now $d(\mathcal{P}, \mathcal{P}')$ is defined as

$$d(\mathcal{P}, \mathcal{P}') = \sum_{x \in N} \bar{\delta}(f(C(x)), C'(x)). \quad (5)$$

It is easy to see that under this definition, f in the first approach is simply the identity function $f(c) = c$ for every $c \in \mathcal{C}$. This essentially defines a *recoloring problem* [20] where the goal is to recolor the least number of elements in \mathcal{P}' (or C') such that $f(C(x)) = C'(x)$ for every element. Hence the cost of f is the number of elements x s.t. $f(C(x)) \neq C'(x)$.

Figure 2 illustrates the idea. In the left side, most of the genes chose their original PM (color) and therefore the identity similarity is the optimal mapping. However, in the right side, most of the blue genes chose the green PM, most of the red genes chose the blue PM, and most of the green genes chose the red PM. Therefore this is also the optimal mapping.

Now, since the mapping is from \mathcal{C} to \mathcal{C} , f is a bijection or simply a *matching* between the set of colors.



In [21] Gusfield noted that the partition distance problem can be casted as an *assignment problem* [29] and hence be solved by a maximum flow in a bipartite graph in time $O(mn + n^2 \log n)$ [30]. Matching problems are among the most classical and well investigated in theoretical, as well as in practical, computer science [31]. Although it has a polynomial time exact algorithms with many flavors [30], a host of works on approximated solutions were introduced. For its very simple implementation and empirically accurate results that are based on theoretical properties we show below, we chose to use a very simple greedy algorithm, named *Greedy PartDist*. The algorithm works recursively and, at each recursion, chooses the heaviest edge (u, v) in the graph, adds it to the matching M and removes from the graph all other edges (u, v') and (u', v) for $u', v' \in V$. The algorithm is given below.

Greedy PartDist $(\mathcal{P}, \mathcal{P}')$:

- Construct the bipartite graph $B(\mathcal{P}, \mathcal{P}', E, w)$ with $w(p, p') = s(p, p')$ for every $p \in \mathcal{P}$ and $p' \in \mathcal{P}'$ s.t. $s(p, p') > 0$
- sort E according to w by descending order and use it as a stack to fetch elements.
- $M_G \leftarrow \emptyset$
- while E is not empty
 - $(p_0, p'_0) \leftarrow pop(E)$
 - $M_G \leftarrow M_G \cup (p_0, p'_0)$
 - Remove all (p_0, p_j) and (p_i, p'_0) items from E
- match arbitrarily zero-degree elements from \mathcal{P} with zero-degree elements from \mathcal{P}' and add to M_G
- Return M_G

Claim. The algorithm *Greedy PartDist* terminates in time $O(m \log n)$.

Proof. The algorithm is comprised from the following major tasks:

- **building the graph B** - Every part $p' \in \mathcal{P}'$ holds the number of elements it has from every $p \in \mathcal{P}$, hence in $O(E)$ B is constructed.
- **Sorting E** - $O(m \log m) = O(m \log n^2) = O(m \log n)$ by any standard sorting procedure.
- **Constructing M_G** - if we go downward on E and additionally holding an auxiliary link from a part to all its edges, each element in E is accessed only once and in constant time, $O(m)$ in total. The final stage consists of matching orphan parts from both partitions. It can be easily perceived that this can be done in time linear in n . □

This algorithm provides a 1/2-approximation guarantee [32]. In the Appendix we provide the same approximation guarantee (i.e. 1/2) by the generic recursive analysis of the *local ratio* technique [33].

The greedy algorithm under the stochastic models

It is interesting to analyze the performance of the greedy algorithm under our stochastic model. It is easy to see (even simply for symmetry arguments) that under our model assumption, every gene remains in its part with probability α (that depends on the two variances σ_P and σ_G) and with probability $1 - \alpha$ chooses uniformly a partition (including its own partition). The expected identity similarity here is the sum over the elements maintaining their part plus those randomly chose that same (original) part.

We therefore obtain:

$$E[s_{\alpha-u}(\mathcal{P}, \mathcal{P}')] = \frac{(k-1)\alpha n}{k} + \frac{n}{k} \tag{6}$$

Definition 1. We say that a PM P is correctly clustered if most of the genes associated with P as a source PM, choose P as their target PM.

Definition 1 is demonstrated visually in Figure 2(a) where every color is preserved (in the right side) by most of its vertices. Definition 1 implies that under a correctly clustered PM, a significant core set of genes stay together is the target PM (part). It is easy to see that, under our stochastic model, if enough genes are associated with every source PM, then all PMs are correctly clustered.

Claim. Assume every color is correctly clustered. Then Algorithm *Greedy PartDist* returns the correct result.

Proof. The proof follows by induction on the number of PMs $|\mathcal{P}|$. For a single PM, there is a single edge in the bipartite graph and this edge is chosen. For $|\mathcal{P}| > 1$, note that by the assumption, the heaviest edge emanating from each PM (node) in \mathcal{P} to its corresponding color in the partition \mathcal{P}' . In particular, this is true for the heaviest edge in the bipartite graph, linking between the nodes corresponding to some PM P . Then the algorithm chooses that edge and remove all edges adjacent to it. Therefore, PM P was correctly chosen and by the induction hypothesis the algorithm returns the correct result. \square

Simulation results

To assess the effectiveness of our PM partitioning identification procedure *PMPI*, we conducted the following simulation study. Number of genes n was held constant $n = 100$ giving rise to $\binom{100}{2} = 4950$ pairs of correlation tests. The number of edges per a gene tree was set to 25, reflecting the average size of the agreement tree among our real data trees. To simulate low agreement similarly to our real data (low MAST value) we discarded every pair with probability $2/3$ maintaining approximately $1/3$ of the pairs (see more details in Section ‘Results on real data’). Every PM P_k was associated with an intrinsic variance σ_P^2 that sets its relative pace to $e^{\beta_{k,j}}$ where $\beta_{k,j} \sim N(0, \sigma_P^2)$. Similarly, every gene sets its rate at period j to $r_{ij} = r_i e^{\alpha_{ij}} e^{\beta_{k,j}}$ where $\alpha_{ij} \sim N(0, \sigma_G^2)$ (See Model Section 1 for full details).

Every gene was associated with a source PM, same number of genes for each PM. Number of PMs k varied from 2 to 10 (i.e. 10 to 50 genes per PM). Distance between genes was set as $1 - r$ from the regression line where the latter was derived by the Deming regression.

This has defined our correlation graph described above.

In order to apply clustering algorithms on the elements, the elements need to be embedded in some Euclidean space. Multidimensional scaling takes a set of dissimilarities (over a set of elements) and returns a set of points in a Euclidean space, such that the distances between the points are approximately equal to the dissimilarities. A set of Euclidean distances on n points can be represented exactly in at most $n - 1$ dimensions. The procedure *cmdscale* follows the analysis of Mardia [34], and returns the best-fitting k -dimensional representation, where k may be less than the argument k (and by definition smaller than n). In our implementation, in order to avoid any distortion, we set k to the maximum value as determined by the data (and is found and returned by the method). We used a version of *cmdscale* that is implemented in R. As *cmdscale* requires a complete graph, we used the min-hop-min-weight (MHMW) algorithm. The output of the MHMW is a complete graph where the weight between any two points is the lightest (min weight) path among all

min hop reliable paths (paths between trees for which correlation was derived). At this point we can use *cmdscale* to map this graph to the Euclidean space. Note however, that this mapping corresponds *not* to the original graph, rather to some approximation of it derived by the output of the MHMW algorithm. This mapping however serves as an initial guess to the iterative mapping of the original, partial, distance matrix. This iterative process is done by the function *isoMDS* implemented in R. This mapping will serve us for the clustering operation. Now, as opposed to real data, here we know the original number of clusters, we can just set this as the number of clusters required. We used *kmeans* implemented by R to obtain the optimal clustering.

Our results appear in Figure 3. The measured quantity is (normalized) partition distance as measured by our *greedy PartDist*. The independent variable is the ratio between σ_G and σ_P . The larger σ_P the more dispersed are the PMs and hence farther from one another. Equivalently, the smaller σ_G , the more concentrated around their PM are the genes. Therefore, we expect that the smaller the ratio σ_G/σ_P is, i.e. PMs are spaced away from each other while their associated genes are more concentrated, we get better results in the sense that more genes remain in their original cluster and successfully identified. Also, we expect that the larger the number of PMs, the greater the mixing between them with genes end up in PMs that are neighboring to their original PMs. Indeed it can be seen that for two and four PMs, for any ratio of $\sigma_G/\sigma_P \leq 1$ a very accurate reconstruction is achieved and so as to six clusters, but for ratio a little less than 1. It is also shown that for every number of PMs, at some critical σ_G/σ_P ratio (that depends on #PMs) the reconstruction curve reaches a saturation that tends to the random similarity as we computed above.

Results on real data

Working with real data poses some other serious problems requiring solution. The first, is that we don’t have here exactly τ periods with edge length ℓ_{ij} for every gene g_i rather a set of trees with loose pairwise agreement. This loose agreement is due to vast discordance between the histories of the various genes as a result of phenomena such as horizontal gene transfer (HGT) or incomplete lineage sorting (ILS, see more details below). However, discordance can arise even from the simple fact that some gene is missing in some specific species, resulting in a contraction of internal nodes. Figure 4 illustrates HGT and missing taxa cases. In both cases, the taxa set $\{1, 2, 4\}$ is an agreement subset for both gene-trees and in this case also maximal agreement subset.

To cope with this problem, we employ the idea of Maximum Agreement Subtrees (MAST) [35], that seeks for the largest subset of species under which the two trees are the

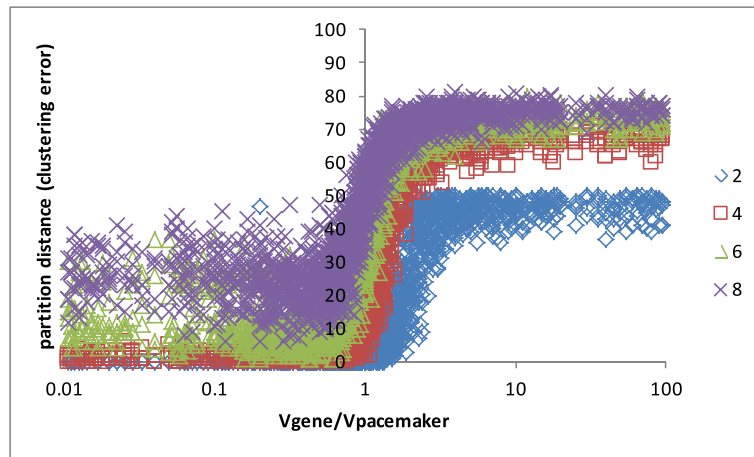


Figure 3 Partition distance obtained by applying the PMPI technique on simulated data versus the gene/pacemaker variance ratio; the plots are shown for 2, 4, 6 and 8 clusters (PMs).

same. Under MAST (or in general, any subset of the leaf set), edges not connecting any species to the induced tree, are removed, and internal nodes with degree two are contracted, while maintaining the original length of the path. Hence for every pair of genes (trees) we need to find the MAST and compare lengths of corresponding edges.

Additionally, here as opposed to a simulation study, we do not know the “real” partition and cannot compare the resultant clustering to it. Therefore, another method for assessing the results should be employed. Here we need to compare the result to the probability of being obtained under a random model. Recall that at the final stage of the PMPI procedure we employ the *kmeans* algorithm which seeks to minimize some *error measure* W_K . This error measure holds the sum of all pairwise distances between members of the same cluster, across all clusters in the partition. It is clear that the more clusters, the smaller W_K is. However, the decrease in W_K is the largest near the

real value of the number of clusters $k = K$, and vanishes slowly for $k > K$. Therefore, a threshold for the improvement (decrease) in W_K must be defined as a stopping condition, above which we don’t increase the number of clusters k . The *gap statistics analysis* [22] compares the improvement in W_K under the real data, to that of a random model. The gap (between the improvements) forms an “elbow” at the optimal (real) K and this is the stopping condition.

The real data we chose to analyze is the one used by us [1] previously, of a set of gene trees that covers 2755 orthologous families from 100 prokaryotic genomes [36]. Prokaryotic evolution is characterized by the pervasive phenomena of horizontal gene transfer (HGT) [37,38], resulting in different topologies for almost any two gene trees. To account for this we employed the MAST procedure for every gene pair and considered this pair only if the MAST contained at least 10 leaves (species).

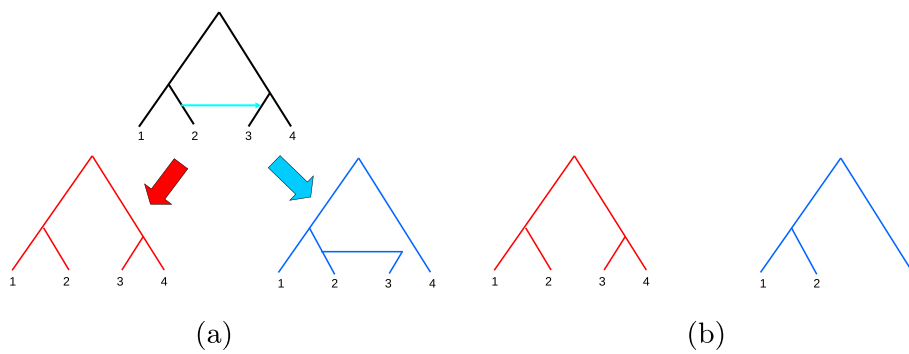


Figure 4 Cases in which agreement subtrees must be taken. **(a)** HGT case: An HGT event occurred in the blue gene, from the edge leading to 2 to the one leading to 3. While the red gene is unaffected, the blue gene has 2 and 3 as sister taxa. **(b)** Missing taxa case: Taxon 3 is missing the blue gene. In both cases (a) and (b), a tree over taxa {1, 2, 4} has the same topology.

Branch lengths of the original trees were used to compute the branch lengths of the corresponding MAST components (by computing path lengths). The variant of Deming regression in the log space as described in Section ‘Simulation analysis’ was performed on the logarithms of the lengths of equivalent branches in both MAST components. The standard sample Pearson correlation coefficient was used as the measure of correlation between the branch lengths. The graph of correlations between the gene trees contained a giant connected component containing 2755 genes and 1,250,972 edges, 33% of the maximum possible number (an edge in the graph exists only when the MAST for the corresponding pair of trees consists of at least 10 species). To cluster these genes according to the correlation between their branch lengths, the data were projected using isoMDS into a 30-dimensional space based on the sparse matrix where $1 - r$ (correlation coefficient) was used as a distance. We ran *k-means* for k spanning the range from 2 to 30. The random model we chose to consider is the fully random uniform model (i.e., $\alpha = 0$, no advantage to source PM) and we compared the results to this model. Grouping these 2755 genes in two clusters containing 1550 and 1205 members, respectively, yields the optimal partitioning according to the gap function statistics (Figure 5). We see the typical “elbow” at the value of $k = 2$. The absolute results were 5,587,960 for the total graph weight, 2,686,914 and 2,285,921 weight within each of the clusters, and 615,125 between them. Analysis of the cluster membership reveals small albeit significant differences in the representation of functional categories of genes but no outstanding biologically relevant trends were detected. Therefore, we can hypothesize

that if indeed the data gives rise to multiple PMs, this signal is completely obscured by the amount of noise produced by the genes themselves (i.e. loose adherence to the associated PM), and noise introduced by artificial factors such as MAST, multiple sequence alignment, and phylogenetic reconstruction.

Conclusions

The universal pacemaker (UPM) model provides a more general framework to analyze genome evolution than the MC model as it makes no assumptions of the absolute evolutionary rates of gene, only on the relative rates. This provides a better explanation to the data observed at extant species. However, similarly to the MC, the UPM is extremely over-dispersed, with the noise complicating detailed analysis. The difficulty in PM analysis is caused both by the weak informative signal and by the large volume of the data.

A natural expectation, however, is for different gene groups, to adhere to different PMs, characterized by different functions. This classification imposes a partition over the gene set where each gene is associated with its own PM. The inference of such a partition is challenging twofold; first from information perspective, as it needs to overcome a high level of “noise”, both biological, as well as artificial. Next, the computational task of solving the PMPI problem requires investigating all possible partitions over the gene set.

In this work we provide the first heuristic procedure for detecting such a partitioning that is based on theoretical ground. We use the Deming regression to infer correlation between pairs of genes, and represent this correlation

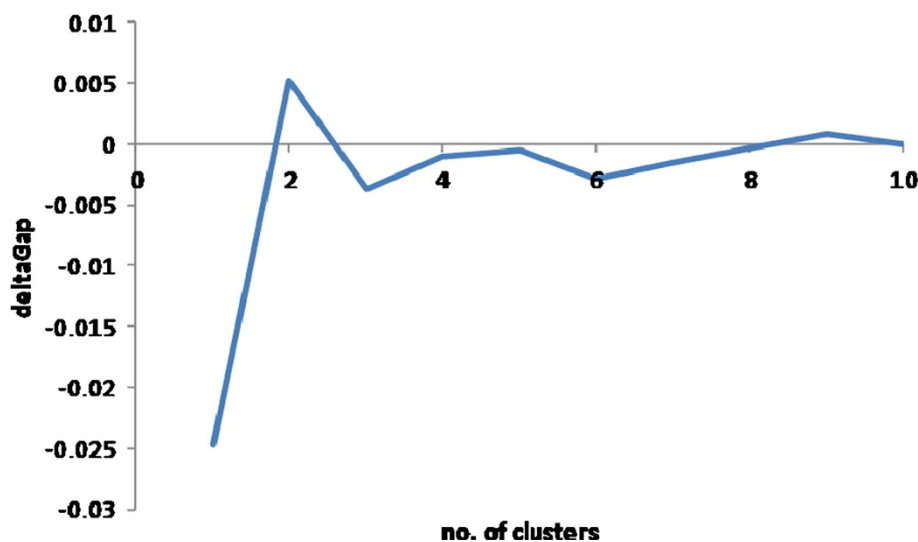


Figure 5 The deltaGap function for 2755 analyzed genes, k from 1 to 10. According to Tibshirani et al. [22], the smallest k producing a non-negative value of $\text{deltaGap}[k] = \text{Gap}[k] - \text{Gap}[k+1] + \sigma[k+1]$ indicates the optimal number of clusters.

relationship in a graph. Subsequently, we embed this graph in the Euclidean space and apply a clustering procedure to it.

We also provide simulation and empirical results of the application of this procedure. In the simulation study, we have shown that the proposed procedure is sound and is capable of detecting the original partition with high accuracy for a fairly small (up to 6) number of PMs as long as the intrinsic gene rate variance is at the size of the PM variance. In the real data realm, we succeed in showing that the analyzed genome-wide set of gene trees is optimally partitioned between two PMs, and the improvement in the statistical explanation is small albeit highly significant. The partition of different functional gene groups between the two PMs is also statistically significant (WRT random partitioning of each group) however the biological interpretation of this partitioning is challenging and remained for future research.

Appendix

Performance guarantees for greedy PartDist algorithm

We now prove a performance guarantee on the output matching returned by the *Greedy PartDist* algorithm. As noted above, it was already shown [32] that the algorithm provides a 1/2-approximation, however the approach used here is different and of separate interest. The first result applies to a general input and hence should be compared to the optimal solution for such cases that is obtained by solving the assignment problem.

A 1/2-approximation maximum weight perfect matching via local ratio

Our approximation algorithm makes use of the local ratio technique, which is useful for approximating optimization covering problems such as vertex cover, dominating set, minimum spanning tree, feedback vertex set and more [33,40,41]. We hereafter describe it briefly, in the context of our setting: The input to the problem is a triplet $(B = (\mathcal{P}, \mathcal{P}', E), f : 2^E \rightarrow \{0, 1\}, w : E \rightarrow \mathbb{R}^+)$ where $B = (\mathcal{P}, \mathcal{P}', E)$ is a bipartite graph, f maps (edge subsets) to the set of valid solutions and let $\Sigma = \{E' \subseteq E : f(E') = 1\}$ be that set. The goal is to find a subset $X \in \Sigma$ such that $w(X)$ is maximized, i.e. $w(X) = OPT(V, \Sigma, w) = \max_{Y \in \Sigma} w(Y)$ (in our context E is the set of edges, and Σ is the set of perfect matchings in B). The local ratio principle is based on the following observation (see e.g. [33]):

Observation 4. For a maximization problem π and every two weight functions w_1, w_2 :

$$OPT_{\pi}(E, \Sigma, w_1) + OPT_{\pi}(E, \Sigma, w_2) \geq OPT_{\pi}(E, \Sigma, w_1 + w_2)$$

Now, given our initial weight function w , we select w_1, w_2 s.t. $w_1 + w_2 = w$ and $|support(w_1)| < |support(w)|$

where $support(w) = \{i : w(i) > 0\}$ (i.e. w_1 is relevant to a smaller set). We first apply the algorithm to find an r -approximation to (E, Σ, w_1) (in particular, if $support(w_1) = \emptyset$, then every matching is an optimal matching to (E, Σ, w_1)). Let X_1 be the solution returned for (E, Σ, w_1) , and assume that $w(X_1) \geq r \cdot OPT(E, \Sigma, w_1)$. If we could also guarantee that $w(X_2) \geq r \cdot OPT(E, \Sigma, w_2)$ then by Observation 4 we are guaranteed that $X = X_1 \cup X_2$ is also an r -approximation for $(E, \Sigma, w_1 + w_2 = w)$. In order to guarantee this we first need to care that the solutions X_1 and X_2 comply. That is, the joint solution composed by X_1 and X_2 is a valid solution (in the simple, unweighted case, this is just a union of X_1 and X_2). We therefore devise the following decomposition of w to w_1 and w_2 :

Definition 2. For a weighted bipartite graph $B = (\mathcal{P}, \mathcal{P}', E, w)$ let $e_M = (p_M, p'_M) \in E$ be the maximal weight edge. Then set

$$w_2(e) = \begin{cases} w(e) & \text{if } e = (p_M, p_j) \text{ or } e = (p_i, p'_M) \text{ for every } p_i \in \mathcal{P} \text{ and } p_j \in \mathcal{P}' \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

That is, w_2 leaves the weight of all edges adjacent to either p_M or p'_M intact, and sets to zero all other edges in $support(w)$. We also let $w_1(e) \leftarrow w(e) - w_2(e)$.

We now analyze the algorithm *Greedy PartDist* above. We prove this by two observations.

Observation 5. Let w_2 be as defined above. Then, at every recursion, the algorithm *Greedy PartDist* provides a $\frac{1}{2}$ -approximation to $support(w_2)$.

Proof. Since at most two edges from $support(w_2)$ are present in an optimal matching M^* , by taking the heaviest edge $e_M = (p_M, p'_M)$ the observation follows. \square

Observation 6. Let X be the matching returned by the algorithm upon return from the recursion, i.e. X is a matching for $B = (\mathcal{P}, \mathcal{P}', E, w_1)$ and assume X is at least a $\frac{1}{2}$ approximation for $support(w_1)$. Then $X \cup e_M = (p_M, p'_M)$ is a valid matching with weight at least $w(M^*)/2$.

Proof. The first part of the claim follows as the set of edges are disjoint and cover all nodes in $support(w)$. The guarantee on the approximation follows by a simple induction on the number of iterations. The basis is when $support(w) = \emptyset$. The induction step follows easily using Observation 5. \square

Competing interests

The authors declare that they have no competing interests.

Acknowledgments

Research was supported in part by the USA-Israel Binational Science Foundation. Part of this work was done while the author was visiting the National Center for Biotechnology Information (NCBI) at the National Institutes of Health (NIH), USA. We thank Eugene Koonin and Yuri Wolf for helpful discussions, in particular in interpretation of the biological significance of the resulted clustering of the real data in Section 'Results on real data'.

An extended abstract has appeared in [39].

Received: 21 October 2014 Accepted: 11 November 2014

Published online: 31 December 2014

References

- Snir S, Wolf YI, Koonin EV: **Universal pacemaker of genome evolution.** *PLoS Comput Biol* 2012, **8**:e1002785, 11.
- Zuckerandl E: **On the molecular evolutionary clock.** *J Mol Evol* 1987, **26**(1):34–46.
- Bromham L, Penny D: **The modern molecular clock.** *Nat Rev Genet* 2003, **4**(3):216–224. doi:10.1038/nrg1020.
- Linder M, Britton T, Sennblad B: **Evaluation of bayesian models of substitution rate evolution parental guidance versus mutual independence.** *Syst Biol* 2011, **60**(3):329–342.
- Allan Drummond D, Wilke CO: **Mistranslation-induced protein misfolding as a dominant constraint on coding-sequence evolution.** *Cell* 2008, **134**(2):341–352.
- Grishin NV, Wolf YI, Koonin EV: **From complete genomes to measures of substitution rate variability within and between proteins.** *Genome Res* 2000, **10**(7):991–1000. doi:10.1101/gr.10.7.991.
- Wolf YI, Novichkov PS, Karev GP, Koonin EV, Lipman DJ: **The universal distribution of evolutionary rates of genes and distinct characteristics of eukaryotic genes of different apparent ages.** *Proc Natl Acad Sci* 2009, **106**(18):7273–7280.
- Bromham L: **Why do species vary in their rate of molecular evolution?** *Biol Lett* 2009, **5**(3):401–404.
- Snir S, Wolf YI, Koonin EV: **Universal pacemaker of genome evolution in animals and fungi and variation of evolutionary rates in diverse organisms.** *Genome Biol Evol* 2014, **6**(6):1268–1278.
- Wolf YI, Snir S, Koonin EV: **Stability along with extreme variability in core genome evolution.** *Genome Biol Evol* 2013, **5**(7):1393–1402.
- Ho SYW: **The changing face of the molecular evolutionary clock.** *Trends in Ecol Evol* 2014, **29**(9):496–503.
- Duchêne S, Molak M, Ho SYW: **Clockstar: choosing the number of relaxed-clock models in molecular phylogenetic analysis.** *Bioinformatics* 2014, **30**(7):1017–1019.
- Ho SYW, Lanfear R: **Improved characterisation of among-lineage rate variation in cetacean mitogenomes using codon-partitioned relaxed clocks.** *Mitochondrial DNA* 2010, **21**:138–146.
- Lanfear R, Calcott B, Ho SYW, Guindon S: **Partitionfinder: Combined selection of partitioning schemes and substitution models for phylogenetic analyses.** *Mol Biol Evol* 2012, **29**(6):1695–1701.
- Wasserman L: *All of Statistics.* New York: Springer; 2004.
- Adcock RJ: **A problem in least squares.** *Ann Mathematics* 1878, **5**:53–54.
- Deming WE: *Statistical Adjustment of Data.* John Wiley & Sons; 1943.
- Fuller WA: *Measurement Error Models.* John Wiley & Sons; 1987.
- Moran S, Snir S: **Efficient approximation of convex recolorings.** *J Comput Syst Sci* 2007, **73**(7):1078–1089.
- Moran S, Snir S: **Convex recolorings of strings and trees: Definitions, hardness results and algorithms.** *J Comput Syst Sci* 2008, **74**(5):850–869.
- Gusfield D: **Partition-distance: A problem and class of perfect graphs arising in clustering.** *Inf Process Lett* 2002, **82**(3):159–164.
- Tibshirani R, Walther G, Hastie T: **Estimating the number of clusters in a data set via the gap statistic.** *J R Stat Soc: Ser B (Stat Methodology)* 2001, **63**(2):411–423.
- Hartigan JA, Wong MA: **A k-means clustering algorithm.** *Appl Stat* 1979, **28**:100–108.
- Lloyd SP: **Least squares quantization in pcm.** *IEEE Trans Inf Theory* 1982, **28**:129–137.
- Kruskal JB, Wish M: *Multidimensional Scaling.* Beverly Hills and London: Sage Publications; 1978.
- Kruskal JB: **Nonmetric multidimensional scaling: a numerical method.** *Psychometrika* 1964, **29**:115–130.
- Guérin R, Orda A: **Computing shortest paths for any number of hops.** *IEEE/ACM Trans Netw* 2002, **10**(5):613–620.
- Borg I, Groenen P: *Modern Multidimensional Scaling, Theory and Applications.* New York: Springer - Verlag; 1997.
- Lawler EL: *Combinatorial Optimization: Networks and Matroids.* The University of Michigan; 1976.
- Ahuja RK, Magnanti TL, Orlin JB: *Network Flows.* Prentice Hall: Englewood Cliffs: New Jersey; 1993.
- Tutte WT: *Connectivity in Graphs.* Mathematical expositions. University of Toronto Press; 1966.
- Preis R: **Linear time 1/2-approximation algorithm for maximum weighted matching in general graphs.** In *STACS 99, 16th Annual Symposium on Theoretical Aspects of Computer Science, March 4-6, 1999, Proceedings*, volume 1563 of *Lecture Notes in Computer Science.* Trier, Germany: Springer; 1999:259–269.
- Bar-Yehuda R: **One for the price of two: a unified approach for approximating covering problems.** *Algorithmica* 2000, **27**:131–144.
- Mardia KV: **Some properties of classical multidimensional scaling.** *Commun Stat - Theory Methods* 1978, **A7**:123–341.
- Finden CR, Gordon AD: **Obtaining common pruned trees.** *J Classification* 1985, **2**:225–276.
- Puigbo P, Wolf Y, Koonin E: **Search for a 'tree of life' in the thicket of the phylogenetic forest.** *J Biol* 2009, **8**(6):59.
- Doolittle WF: **Phylogenetic classification and the universal tree.** *Science* 1999, **284**(5423):2124–9.
- Gogarten JP, Doolittle WF, Lawrence JG: **Prokaryotic evolution in light of gene transfer.** *Mol Biol Evol* 2002, **19**:2226–2238.
- Snir S: **Pacemaker partition identification.** In *Algorithms in Bioinformatics - 14th International Workshop, WABI 2014, September 8–10, 2014, Proceedings*, volume 8701 of *Lecture Notes in Computer Science*, 2014: 281–295. Springer: Wroclaw, Poland; 2014.
- Bafna V, Berman P, Fujito T: **A 2-approximation algorithm for the undirected feedback vertex set problem.** *SIAM J Discrete Math* 1999, **12**:289–297.
- Bar-Yehuda R, Even S: **A local-ratio theorem for approximating the weighted vertex cover problem.** *Ann Discrete Math* 1985, **25**:27–46.

Submit your next manuscript to BioMed Central and take full advantage of:

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

