Identifying novel constrained elements by exploiting biased substitution patterns

Manuel Garber[1] , Mitchell Guttman[1] , Michele Clamp[1] , Michael C Zody[1,2] , Nir Friedman[3] and Xiaohui Xie[1,4*]

[1]Broad Institute of MIT and Harvard, 7 Cambridge Center, Cambridge, MA 02142, USA
[2]Department of Medical Biochemistry and Microbiology, Uppsala University, Uppsala, Sweden
[3]School of Computer Science and Engineering, Hebrew University, Jerusalem 91904, Israel
[4]Department of Computer Science, University of California, Irvine, CA 92697, USA

Email: Xiaohui Xie - xhx@ics.uci.edu;

*Corresponding author

1 Supplementary Methods

1.1 Probabilistic model of phylogeny

Suppose we are provided with a set of aligned sequences from M species, whose evolutionary relationship is described by a phylogenetic tree $\mathcal{T} = (\mathcal{N}, \mathcal{E})$, specified by a collection of nodes $\mathcal{N}$ and edges $\mathcal{E}$ (see for example Supp Figure 1). Assume $\mathcal{T}$ is a rooted binary tree, in which case the tree contains $N = 2M - 1$ nodes with $M - 1$ being internal (corresponding to ancestral species) and not directly observable. For simplicity, we index the leaf nodes from 1 to M, and the ancestral nodes from $M + 1$ to N, always using N for the root node.

Consider *one column* of aligned sequences among the M species. Denote the aligned sequence by $(x_1, x_2, \ldots, x_M)$ with x_i representing the sequence from species i, and assume x_i is drawn from a finite set of states, i.e. $x_i \in \mathcal{A} = \{a_1, a_2, \ldots, a_K\}$. In the context of nucleotide sequence, $K = 4$ and x_i belongs to one of four nucleotides, that is $\mathcal{A} = \{A, C, G, T\}$.

We assume the M aligned nucleotides are described by a probabilistic model with conditional dependency specified by the phylogenetic tree $\mathcal{T}$. Denote the state variable at node i by x_i for all

$i = 1, \cdots, N$, of which only the variables $(x_1, \cdots, x_M)$ at the leaf nodes are directly observable. Denote the parent of node i by $\text{pa}(i)$ and the probability of x_i conditioned on its parent by $P(x_i|x_{\text{pa}(i)})$. The joint probability of all variables in the tree is then described by

$$P(x_1, x_2, \ldots, x_N|\mathcal{T}) = P(x_N) \prod_{i=1}^{N-1} P(x_i|x_{\text{pa}(i)}) \tag{1}$$

where $P(x_N)$ represents the prior probability of the root node variable.

1.1.1 Modeling molecular evolution

We use a continuous-time Markov process (CTMP) to model the evolution of nucleotide sequence from $X(0)$ at time 0 to $X(t)$ at time t. We assume the process is homogeneous and the transition rate matrix between different states is specified by $\mathbf{Q}$. The probability of observing b at time t conditioned on the starting state being a is then given by

$$P(X(t) = b|X(0) = a) = \left[e^{Qt}\right]_{ab} \tag{2}$$

for all $a, b \in \mathcal{A}$. $[\cdot]_{ab}$ represents the entry of the matrix at row $I(a)$ and column $I(b)$, with $I(a)$ denoting the index of the state a in the alphabet $\mathcal{A}$. Under this model, the conditional probability in Eq. (1) is specified by $P(x_i = b|x_{\text{pa}(i)} = a) = \left[e^{Qt_i}\right]_{ab}$, where t_i is the length of the edge leading to node i.

The CTMP process is said to be time-reversible if

$$P(X(t) = b|X(0) = a) = P(X(0) = b|X(t) = a) \tag{3}$$

for any a, b or t. For reversible CTMP, there exists a vector π such that

$$\pi_i Q_{ij} = \pi_j Q_{ji} \tag{4}$$

for all i and j. In this case, the vector π is also the stationary distribution of the CTMP, that is, $\pi' \exp(\mathbf{Q}t) = \pi'$ (or $\pi'\mathbf{Q} = 0$). Furthermore, it can be shown that the CTMP with rate matrix $\mathbf{Q}$ is time-reversible if and only if $\mathbf{Q}$ can be decomposed as

$$Q_{ij} = \pi_j R_{ij} \text{ or written in the matrix form } \mathbf{Q} = \mathbf{RD}(\pi) \tag{5}$$

where $\mathbf{R} = \mathbf{R}'$ is symmetric, and $\mathbf{D}(\pi) = \text{diag}(\pi)$ is a diagonal matrix with elements specified by π. In other words, the evolutionary model (Eq. (19)) we use in the paper is time-reversible.

Calculation of the transitional matrix (Eq. (2)) can be greatly simplified if the matrix $\mathbf{Q}$ is digonalizable. Under this assumption, $\mathbf{Q}$ can be factorized as $\mathbf{Q} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-\mathbf{1}}$, where $\mathbf{U}$ is the eigenvector matrix and $\mathbf{\Lambda}$ is the diagonal matrix with eigenvalues of $\mathbf{Q}$. Consequently, Eq. (2) can be expressed as

$$P(X(t)|X(0)) = \exp(\mathbf{Q}t) = \mathbf{U}\exp(\mathbf{\Lambda})\mathbf{U}^{-1} \tag{6}$$

In general, the eigenvector $\mathbf{U}$ and eigenvalue matrix and $\mathbf{\Lambda}$ may not be real. However, for the subset of transition rate matrices that are time-reversible (Eq. (5)), both $\mathbf{U}$ and $\mathbf{\Lambda}$ are always real, which greatly simplifies the calculation of the conditional probability Eq. (2).

1.2 Inference

Note that in the probabilistic model Eq. (1), only the variables in the M leaf nodes are directly observable. All variables in the ancestral nodes have to be inferred. In particular, we are interested in the following three inference problems:

1) The likelihood of observing the aligned nucleotides, $x^{\text{obs}} = (x_1, x_2, \cdots, x_M)$, in the leaf nodes

$$\mathcal{L}(x^{\text{obs}}|\mathcal{T}) = P(x^{\text{obs}}|\mathcal{T}) = \sum_{x_{M+1},\ldots,x_N} P(x_1, \ldots, x_M, x_{M+1}, \ldots, x_N|\mathcal{T}) \tag{7}$$

2) Posterior probability of observing b in node i and a in its parent node $\text{pa}(i)$

$$q_{ab}^i = P(x_{\text{pa}(i)} = a, x_i = b|x^{\text{obs}}, \mathcal{T}) \tag{8}$$

3) Posterior probability of observing a in node i

$$q_a^i = P(x_i = a|x^{\text{obs}}, \mathcal{T}) \tag{9}$$

Because of the inherent tree structure relating them, these variables can be decoupled and the summation can be done in linear time. This is done using Felsenstein's pruning and peeling

algorithm, which uses dynamic programming and is also referred to as sum-product rule or belief propagation in probabilistic models.

The Felsenstein algorithm first collects message (evidence) from leaf nodes. After this is done, it then distributes collected messages from the root node to leaf nodes. The two steps are shown in the following (see details in [1, 2]).

1.2.1 *Message collection or upward belief propagation*

Denote $\mathcal{T}_n$ the substree of node n, which consists of node n and all the descendants of node n. Let $\alpha^n(a)$ be the likelihood of observing the given sequences at the leaf nodes of $\mathcal{T}_n$ conditioned on $x_n = a$, that is,

$$\alpha^n(a) = P(\{x_i|\text{for all node } i \text{ that is a leaf node in } \mathcal{T}_n\}|x_n = a) \tag{10}$$

It follows that $\alpha^n(a)$ can be calculated recursively starting from leaf nodes:

- If n is a leaf node,

$$\alpha^n(a) = \delta_{a,x_n} \tag{11}$$

- If n is an internal node,

$$\alpha^n(a) = \prod_{\{c|\text{pa}(c)=n\}} \sum_b P(x_c = b|x_n = a)\alpha^c(b) \tag{12}$$

1.2.2 *Message distribution or downward belief propagation*

Denote the complement of the substree $\mathcal{T}_n$ by $\bar{\mathcal{T}}_n$, which includes the parent of n, $\bar{\mathcal{T}}_{\text{pa}(n)}$, and the sister node of n, denoted by $s(n)$. Let $\beta^n(a)$ be the likelihood of observing the given sequences at the leaf nodes of $\bar{\mathcal{T}}_n$ in conjunction with $x_{\text{pa}(n)} = a$, that is,

$$\beta^n(a) = P(\{x_i|\text{for all node } i \text{ that is a leaf node in } \bar{\mathcal{T}}_n\},\ x_{\text{pa}(n)} = a) \tag{13}$$

$\beta^n(a)$ can also be computed recursively, starting from the root node:

- If $\mathrm{pa}(n) = N$ (i.e. $\mathrm{pa}(n)$ is the root node),

$$\beta^n(a) = \sum_{x_{s(n)}} P(x_{s(n)}|x_{\mathrm{pa}(n)} = a)\alpha^{s(n)}(x_{s(n)})P(x_N = a) \tag{14}$$

- Otherwise,

$$\beta^n(a) = \sum_{x_{s(n)}} P(x_{s(n)}|x_{\mathrm{pa}(n)} = a)\alpha^{s(n)}(x_{s(n)}) \sum_c \beta^{\mathrm{pa}(n)}(c)P(x_{\mathrm{pa}(n)} = a|x_{\mathrm{pa}(\mathrm{pa}(n))} = c) \tag{15}$$

1.2.3 Likelihood function and posterior probability

Provided with α and β, the inference problems in Eq. (7), Eq. (8) and Eq. (9) can be solved efficiently. The likelihood of observing M variable in the leaf nodes (aligned nucleotides)

$$\mathcal{L}(x^{\mathrm{obs}}) = \sum_a \alpha^N(a)P(x_N = a) \tag{16}$$

The posterior probability of observing b in node i and a in its parent $\mathrm{pa}(i)$

$$q^i_{ab} = \beta^i(a)P(x_i = b|x_{\mathrm{pa}(i)} = a)\alpha^i(b)/\mathcal{L}(x^{\mathrm{obs}}) \tag{17}$$

And the posterior probability of observing b in node i is

$$q^i_b = \sum_a q^i_{ab} \tag{18}$$

1.3 Learning

In the paper we used the following instantaneous rate matrix $\mathbf{Q}$ for the CTMP

$$Q_{ab} = \omega\pi_b R_{ab} \tag{19}$$

for the transition rate from a to b. Assuming the matrix R is given (it can be estimated using maximum likelihood phylogenetic tools like PAML [3] or the Phast package [4], our goal is to learn the scaling factor ω and the vector π from the observed sequence alignment

$x^{\text{obs}} = (x_1, x_2, \cdots, x_M)$ in M species. We use the maximum likelihood method to estimate ω and π , that is, we find $\hat{\omega}$ and $\hat{\pi}$ that maximizes the log likelihood function

$$(\hat{\omega}, \hat{\pi}) = \underset{\omega,\pi}{\text{argmax}} \ \log \mathcal{L}(x^{\text{obs}}|\omega, R, \pi) \tag{20}$$

1.3.1 EM-algorithm

We use the EM-algorithm ([5]) to solve Eq. (20). The algorithm iterates in the following two steps:

1) **E-step:** Infer the posterior probability of the ancestral sequences using the the parameters leaned in the previous step $\Theta^{(t-1)}$,

$$Q^{(t)}(x_{M+1}, \ldots, x_N) = P(x_{M+1}, \ldots, x_N|x^{\text{obs}}, Q^{(t-1)}) \tag{21}$$

2) **M-step:** Find new parameters that maximize the averaged log likelihood function

$$Q^{(t)} = \text{argmax}_Q \sum_{x_{M+1},\ldots,x_N} Q^{(t)}(x_{M+1}, \ldots, x_N) \log P(x_1, x_2, \cdots, x_N|Q) \tag{22}$$

Because the likelihood function Eq. (1) in the above equation can be factorized into a product form, the maximization problem in the M-step can be simplified to the following form

$$Q^{(t)} = \text{argmax}_Q \ \sum_{i=1}^{N-1} Q^{(t)}(x_i, x_{\text{pa}(i)}) \log P(x_i|x_{\text{pa}(i)}) + Q^{(t)}(x_N) \log P(x_N) \tag{23}$$

where $Q^{(t)}(x_i, x_{\text{pa}(i)})$ is the marginal probability of observing x_i in node i and $x_{\text{pa}(i)}$ in the parent of node i, and $Q^{(t)}(x_N)$ is the marginal probability of observing x_N in the root node. This suggests that in the E-step we only need to calculate the two sets of conditional probabilities, both of which can be efficiently calculated using Eq. (17) and Eq. (18) respectively.

1.3.2 Sufficient statistics of CTMP

The gradient of the averaged log likelihood function in Eq. (23) with respect to Q is in general not easy to calculate because Eq. (2) involves an exponential matrix term. However, the gradient can

be greatly simplified by using the sufficient statistics to summarize the CTMP at each branch of the tree $\mathcal{T}$ [6, 7].

Consider a CTMP starting from state $X(0) = a$ and ending at state $X(t) = b$. The transition probability from a to b is fully specified by the duration of each state $T(k|a, b)$ and the number of transitions between states, $N(k, l|a, b)$ for the number of transitions from state k to l,

$$\log P(X(t) = b|X(0) = a) \sim \sum_k T(k|a, b)Q_{kk} + \sum_k \sum_{l \neq k} N(k, l|a, b) \log Q_{kl} \tag{24}$$

In general, both $T(k|a, b)$ and $N(k, l)|a, b)$ depend on Q. However, we can treat them as latent variables and infer their posterior distributions at the E-step of the EM-algorithm. This results in a reformulation of the averaged log likelihood function in Eq. (23):

$$\tilde{\mathcal{L}}(Q) \sim \sum_k \mathrm{E}[T(k)]Q_{kk} + \sum_k \sum_{l \neq k} \mathrm{E}[N(k, l)] \log Q_{kl} + \sum_k \mathrm{E}\left[\log P(x_N = k)\right] \tag{25}$$

where $\mathrm{E}[T(k)]$ is the expected duration of state k summed over all branches of the tree, and similarly $\mathrm{E}[N(k, l)]$ is the expected total number of transitions from k to l.

1.3.3 Expectation of sufficient statistics

Consider a homogeneous CTMP with rate matrix Q that starts with a at time 0 ($X(0) = a$) and end with b at time t ($X(t) = b$). Denote the transition probability from a to b by $M_{ab}(t)$, that is,

$$P_{ab}(t) \equiv [\exp(\mathbf{Q}t)]_{ab} \tag{26}$$

During the time period t, the expected duration for the state variable being k is

$$T(k|a, b, t) = \int_0^t P_{ak}(\tau)P_{kb}(t - \tau)d\tau / P_{ab}(t) \tag{27}$$

The expected number of transitions from k to l is

$$N(k, l|a, b, t) = \int_0^t P_{ak}(\tau)Q_{kl}P_{lb}(t - \tau)d\tau / P_{ab}(t) \tag{28}$$

After reorganizing terms [7], it follows that

$$T(k|a, b, t) = \Psi(k, k|a, b, t)/P_{ab}(t) \tag{29}$$

$$N(k, l|a, b, t) = \Psi(k, l|a, b, t)Q_{kl}/P_{ab}(t) \tag{30}$$

where

$$\Psi(i,j|a,b,t) \equiv \int_0^t P_{ai}(\tau)P_{jb}(t-\tau)d\tau \tag{31}$$

$$= \sum_k U_{ak}U_{ki}^{-1}\sum_l U_{jl}U_{lb}^{-1}J_{kl} \tag{32}$$

where $\mathbf{U}$ is the eigenvector matrix of the matrix $\mathbf{Q}$ (Eq. (6)), and matrix $\mathbf{J}$ is defined to be

$$J_{ij}(t) \equiv \begin{cases} t\exp(\lambda_i t) & \text{if } \lambda_i = \lambda_j; \\ \frac{\exp(\lambda_i t)-\exp(\lambda_j t)}{\lambda_i-\lambda_j} & o.w. \end{cases} \tag{33}$$

with λ_i being the eigenvalue of $\mathbf{Q}$ corresponding to eigenvector U_i. We have assumed that eigenvalues of $\mathbf{Q}$ are all real, which is satisfied when $\mathbf{Q}$ takes the form of Eq. (5).

Note that the expectations of sufficient statistics in Eqs. (29, 30) depend on the variables in the internal nodes of the tree. If we treat them as latent variables in the E-step of the EM-algorithm, we will need to calculate posterior expectations of the two sufficient statistics, using the marginal probability Eq. (17).

Consider the edge between node k and its parent node $\mathrm{pa}(k)$. The posterior expected number of transitions from i to j that have occurred at the edge is

$$N^k(i,j) = \sum_{a,b} P(x_{\mathrm{pa}(k)} = a, x_k = b)\Psi(i,j|a,b,t_k)Q_{ij}/P_{ab}(t_k) \tag{34}$$

and the posterior expected number of during for the state variable being i is

$$T^k(i) = N^k(i,i)/Q_{ii} \tag{35}$$

Summing over all edges of the tree, we obtain

$$E[N(i,j)] = \sum_k N^k(i,j) \qquad E[T(i)] = \sum_k T^k(i) \tag{36}$$

based on which the averaged log likelihood function Eq. (25) can then be calculated.

1.3.4 Derivation of the update rules in M-step

The derivative of the averaged log likelihood function Eq. (25) with respective to a parameter θ in Q is

$$\frac{\partial \tilde{\mathcal{L}}(Q)}{\partial \theta} = \sum_k E[T(k)] \frac{\partial}{\partial \theta} Q_{kk} + \sum_k \sum_{l \neq k} E[N(k,l)] \frac{\partial}{\partial \theta} \log(Q_{kl}) + \sum_k q_k^N \frac{\partial}{\partial \theta} \log P(x_N = k) \quad (37)$$

The form of the Q matrix in Eq. (19) leads to the following derivatives:

$$\begin{aligned}
\frac{\partial Q_{aa}}{\partial \omega} &= -\sum_b \pi_b R_{ab}(1 - \delta_{ab}) && (38)\\
\frac{\partial Q_{aa}}{\partial \pi_b} &= -(1 - \delta_{ab}) R_{ab} \omega && (39)\\
\frac{\partial \log(Q_{ab})}{\partial \omega} &= \frac{1}{\omega} && (40)\\
\frac{\partial \log(Q_{ab})}{\partial \pi_b} &= \frac{1}{\pi_b} && (41)
\end{aligned}$$

We assume the prior distribution of the variable in the root node is specified by π, i.e. $P(x_N = a) = \pi_a$. Substituting the above derivatives into Eq. (37) and setting the equation to 0 leads to the following update rules for π and ω,

$$\begin{aligned}
\omega &= \frac{\sum_a \sum_{b \neq a} N(a,b)}{\sum_a T(a) \sum_{b \neq a} R_{ab} \pi_b} && (42)\\
\pi_b &= \frac{\sum_{a \neq b} N(a,b) + q_b^N}{\sum_{a \neq b} T(a) R_{ab} \omega + \gamma} && (43)
\end{aligned}$$

where γ is a Lagrange multiplier and can be found by solving $\sum_b \pi_b = 1$. This completes M-step of the EM-algorithm. Note that both ω and π are obstained in one step without using gradient-based methods. This significantly improves the speed of the algorithm.

2 Neutral model

As described in the Methods of the paper we used PAML [3] to estimate branch lengths of the placental phylogeny using a combined ancestral repeat alignment. The estimated tree (Figure 1) had a total branch length of 2.72 substitutions/site, stationary base distribution

$\pi_0 = (0.295631, 0.205573, 0.202829, 0.295967)$ and rate matrix

$$\begin{pmatrix} -0.8578 & 0.1607 & 0.5538 & 0.1432 \\ 0.2312 & -1.1958 & 0.1630 & 0.8015 \\ 0.8072 & 0.1652 & -1.2095 & 0.2371 \\ 0.1430 & 0.5567 & 0.1625 & -0.8623 \end{pmatrix}$$

To estimate regional variation we also used PAML to estimate branch lengths for each regions separately and also computed the mean ω for each of the regions. Both quantities are very well correlated ($r^2 = 0.83$) and although there is a two fold difference between the shortest and and longest total branch length, predictions based on either using regional models for each region or one combined model for all regions did not alter any of the results in the main text.

References

1. Felsenstein J, et al.: *Inferring phylogenies*. Sinauer Associates Sunderland, Mass., USA 2003.
2. Durbin R, Eddy SR, Krogh A, Mitchison G: *Biological Sequence Analysis*. New York: Cambridge University Press 1998.
3. Yang Z: **PAML 4: Phylogenetic Analysis by Maximum Likelihood**. *Molecular Biology and Evolution* 2007, **24**(8).
4. Siepel A, Bejerano G, Pedersen JS, Hinrichs AS, Hou M, Rosenbloom K, Clawson H, Spieth J, Hillier LW, Richards S, Weinstock GM, Wilson RK, Gibbs RA, Kent WJ, Miller W, Haussler D: **Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes**. *Genome Research* 2005, **15**:1034–1050.
5. Dempster A, Laird N, Rubin D, et al.: **Maximum likelihood from incomplete data via the EM algorithm**. *Journal of the Royal Statistical Society* 1977, **39**:1–38.
6. Holmes I, Rubin GM: **An expectation maximization algorithm for training hidden substitution models**. *Journal of Molecular Biology* 2002, **317**(5):753–764.
7. Hobolth A, Jensen J: **Statistical inference in evolutionary models of DNA sequences via the EM algorithm.** *Statistical applications in genetics and molecular biology* 2005, **4**.

Tables

Table 1 - Genomic locations of the elements uniquely identified by SiPhy

First two columns show the comparison between SiPhy and PhastCons, whereas the last two columns show the comparison between SiPhy and GERP. Each column shows the distribution of the elements uniquely found by the corresponding method, in terms of the number of bases in different functional regions. Estimated FDR for the SiPhy elements using ancestral repeats (not shuffled) as a control is 5%.

Region	**SiPhy**	**PhastCons**	**SiPhy**	**GERP**
Coding Exons	36,856(9.3%)	11,139(2.0%)	34,584(9.2%)	15,495(3.8%)
Intronic	164,110(41.6%)	277,239 (49.1%)	153,644(40.9%)	204,935(49.6%)
$5'$ UTR	6,981(1.8%)	329(0.0%)	5,732(1.5%)	1,305(0.3%)
$3'$ UTR	16,322(4.1%)	12,812(2.2%)	13,547(3.6%)	16,828(4.1%)
5kb from TSS	41,772(10.6%)	16,313(2.9%)	36,949(9.8%)	13,324(3.2%)
5kb downstream	17,205(4,4%)	9,205(1.6%)	16,886(4.5%)	14,610(3.5%)
Intergenic	111,649(28.3%)	237,373(38.7%)	114,525(30.5%)	147,969(35.8%)
Total	394,894	564,410	375,867	413,161

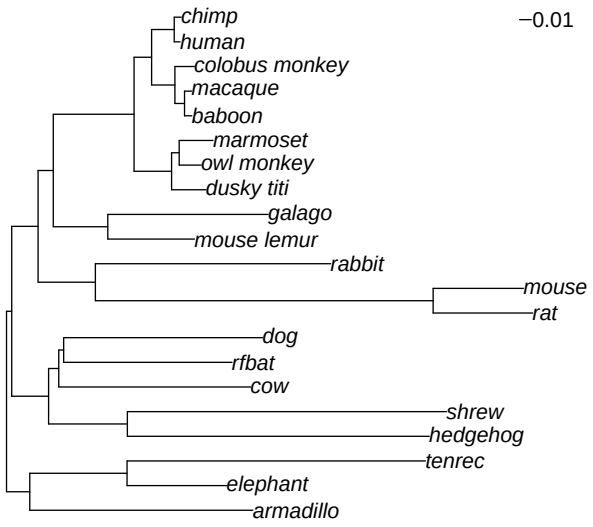

Figure 1: The neutral tree constructed from ancestral repeats in the ENCODE regions is shown. The scale, in substitutions per site, is indicated in the legend.