BMC Evolutionary Biology

Research

Detecting non-coding selective pressure in coding regions

Hui Chen and Mathieu Blanchette*

Address: McGill Centre for Bioinformatics, McGill University, 3775 University St., room 332, Montreal, QC, Canada H3A 2B4

Email: Hui Chen - hui@mcb.mcgill.ca; Mathieu Blanchette* - blanchem@mcb.mcgill.ca

* Corresponding author

from First International Conference on Phylogenomics
Sainte-Adèle, Québec, Canada. 15–19 March, 2006

Published: 8 February 2007

BMC Evolutionary Biology 2007, **7**(Suppl 1):S9 doi:10.1186/1471-2148-7-S1-S9

Abstract

Background: Comparative genomics approaches, where orthologous DNA regions are compared and inter-species conserved regions are identified, have proven extremely powerful for identifying non-coding regulatory regions located in intergenic or intronic regions. However, non-coding functional elements can also be located within coding region, as is common for exonic splicing enhancers, some transcription factor binding sites, and RNA secondary structure elements affecting mRNA stability, localization, or translation. Since these functional elements are located in regions that are themselves highly conserved because they are coding for a protein, they generally escaped detection by comparative genomics approaches.

Results: We introduce a comparative genomics approach for detecting non-coding functional elements located within coding regions. Codon evolution is modeled as a mixture of codon substitution models, where each component of the mixture describes the evolution of codons under a specific type of coding selective pressure. We show how to compute the posterior distribution of the entropy and parsimony scores under this null model of codon evolution. The method is applied to a set of growth hormone 1 orthologous mRNA sequences and a known exonic splicing elements is detected. The analysis of a set of *CORTBP2* orthologous genes reveals a region of several hundred base pairs under strong non-coding selective pressure whose function remains unknown.

Conclusion: Non-coding functional elements, in particular those involved in post-transcriptional regulation, are likely to be much more prevalent than is currently known. With the numerous genome sequencing projects underway, comparative genomics approaches like that proposed here are likely to become increasingly powerful at detecting such elements.

Background

Vertebrate genomes are now recognized as containing a huge number of non-coding functional regions, a large fraction of which is likely to be involved in regulating the various steps of gene expression [1-4]. While most of the attention has been centered on understanding the regulation of transcription, post-transcriptional regulatory mechanisms now appear to be more important than originally thought. Cis-regulation of pre-mRNA splicing is believed to be operated by splicing factors binding intronic and exonic splicing enhancers and helping to include or exclude specific exons from the transcript [5,6].

Post-splicing, parts of the mature mRNA often folds into some RNA secondary structure that determines the level of mRNA degradation [7] as well as mRNA localization [8]. Translational efficiency and accuracy have been shown to be largely determined by the choice of synonymous codon, thus imposing some selective pressure on the codons of certain genes [9]. Translation is also known to be affected by certain secondary structure elements in the mRNA [10]. While most of the known examples of formation of functional secondary structure are restricted to the 5' and 3' UTRs, the coding portion of the mRNA has also been shown to form functional structures [11]. Finally, there are also examples of transcription factor binding sites located in coding exons (e.g. in CD28 [12]). The method presented here should allow the detection of many of these functional elements, which we call coding regions under non-coding selection (CRUNCS). To this point, the computational methods that have proven the most valuable for identifying non-coding functional regions are based on comparative genomics. The guiding principle of this family of approaches is that functional features of a DNA sequence tend to evolve slower than non-functional ones, because of selective pressure. This simple idea is at the core of phylogenetic footprinting, a method that compares orthologous regulatory DNA regions to identify short conserved motifs likely to be transcription factor binding sites [13,14]. The key here is that most of the DNA in promoter regions is non-functional, with the exception of the regulatory elements we are interested in. The same reasoning applies to the detection of intronic splicing enhancers [15]. With the ongoing sequencing of a large number of vertebrate genomes [16], the power of these methods is quickly improving and, coupled with algorithmic improvements [17], they are now able to detect very short regions under selection, or regions under weak selection.

The search for CRUNCS is more challenging. Although the same "conservation implies function" principle applies in this case, it needs to be used more cautiously. Indeed, CRUNCS are *not* located in non-functional sequences as are, for example, most known transcription factors binding sites, but rather in *coding* regions. This means that the sequence conservation observed in exons may be the result of two types of selective pressures. The first one is the pressure to maintain the function of the protein encoded by the gene, which probably explains most of the sequence conservation observed in coding regions. The second type of selective pressure applies only to CRUNCS, which are required to maintain their regulatory role. To apply phylogenetic footprinting to the detection of CRUNCS, one needs to determine which type of selective pressure is responsible for the sequence conservation observed.

The method suggested here takes a conservative approach to the problem. Given a set of aligned orthologous coding sequences, we first evaluate the degree of conservation of each column of the alignment, using either a parsimony score or an entropy score. We then put the burden of explaining the conservation observed as much as possible on the shoulders of the selective pressure on the protein product. Because most amino acids are encoded by many synonymous codons, amino acid selective pressure leaves room for some sequence variation. A region of the sequence will be predicted to be an CRUNCS only if the conservation observed cannot be explained solely by the need for conservation of the encoded amino acids. The method introduced here build a mixture model of codon evolution, and then uses it as a null model to assess the significance of the observed degree of conservation. We illustrate our approach on two sets of orthologous vertebrate genes (growth hormone 1 and *CORTBP2*) and compare it to a related approach by Blanchette [18].

Results and Discussion

Given a multiple alignment of orthologous mRNA sequences, our goal is to identify alignment columns that are conserved beyond what would be expected by chance if the corresponding sites were evolving only under the selective pressure on the amino acid they contribute to encode. Such sites are likely to be under non-coding selective pressure. This section, which constitutes the main contribution of this paper, is structured as follows. First, we define two commonly used sequence conservation scoring methods: the entropy, and the parsimony score. We then describe a methods for assigning a p-value to a given entropy or parsimony score, under null models of evolution of codons that are only under coding selective pressure. Under this method, we model codon evolution as a mixture of codon substitution models and use these models to assign a posterior p-value to a given conservation score.

Two measures of sequence conservation

A number of methods have been proposed to measure the degree of conservation of a set of orthologous sequences and to identify regions under selective pressure (see [19] for an evaluation of some of these methods for finding regulatory elements in intergenic regions). In this paper, we consider two such methods, the entropy score and the parsimony score, and later show how to assess the statistical significance of these scores in the context of coding regions.

Entropy

In the area of transcription factor binding sites detection, a popular method for evaluating sequence conservation is the entropy (see, for example, [20]). Given a set of orthologous nucleotides $x_1, x_2, \ldots, x_n$ from n different species, the

entropy measures the distance between the observed frequency of A, C, G, and T's at that site and the uniform distribution: $entropy\ (x_1, x_2, ...,x_n) = -\sum\alpha\in\{A,C,G,T\}\ f_\alpha \log_2(f_\alpha)$, where f_α is the relative frequency of nucleotide α. Perfectly conserved sites have an entropy of zero, while the worst levels of conservation obtain a score of 2.

Parsimony score

A major drawback of the entropy score is that it does not take into consideration the phylogenetic relationships among the sequences being compared, and indeed the method is mostly used for motif discovery within a single species. An alternative to the entropy score is the parsimony score [21]. Given a set of orthologous nucleotides $x_1, x_2, ..., x_n$ and a phylogenetic tree T whose leaves are labeled with these nucleotides, the parsimony score is defined as the minimum number of substitutions, performed along the branches of the tree T, that can explain the set of nucleotides observed at the leaves. It is thus a lower bound on the actual number of substitutions at that site, and, in cases where this number is not too large compared to the number of branches in the tree, it is a fairly good estimate of the actual number of substitutions that occurred at that site. The parsimony score of a set of n nucleotides can be computed in time $O(n)$ using Sankoff's algorithm [22] or Fitch's algorithm [21]. Figure 1 gives an example of orthologous nucleotides whose conservation is well characterized by either the entropy or the parsimony scores. Entropy and parsimony scores attempt to measure the total selective pressure on a given site. While, for non-coding regions, this selective pressure can be assumed to come completely from the presence of non-coding functional elements, this is not the case in coding regions. The rest of this paper describes how to measure the statistical significance of a certain conservation (entropy or parsimony) score when the site under study lies within a coding region.

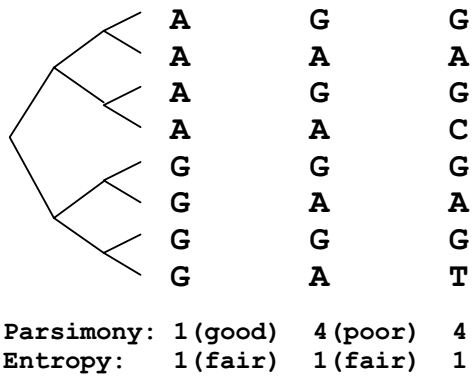

Figure 1
Example of alignment columns where parsimony score and relative entropy differ greatly in their assessment of sequence conservation.

Conservation p-values under a mixture of codon models

In this section, we introduce a null model of coding sequence evolution that consists of a mixture of codon substitution models representing the evolution of codons that are under different types of *purely* coding selective pressure. We then show how to compute posterior p-values for the entropy and parsimony scores of orthologous sequences evolving under those models.

Mixture models for codon evolution

Different positions in a protein sequence are usually subject to different types of coding selective pressures. Some are constrained to have a specific amino acid (e.g. the active site in a zinc finger has to be a cystein), while others are free to have any residues with some particular chemical properties (say, a hydrophobic residue), and still others are under little or no selective pressure at all. Selective pressure on amino acids translates into selective pressure on codons, which explains part of the sequence conservation observed at the mRNA level in coding exons. We describe a mixture model of amino acid evolution, and the corresponding mixture model of codon evolution. We derive a set of 50 amino acid substitution rate matrices $\mathbf{Q}_1^a$, ..., $\mathbf{Q}_{50}^a$, together with codon rate matrices $\mathbf{Q}_1^c$,..., $\mathbf{Q}_{50}^c$, describing the substitution rates for as many classes of selective pressures on amino acids. The choice of modeling codon evolution with only 50 classes is a compromise between an highly accurate modeling of codon evolution (which would most likely require a larger number of classes [23]) and constraints on computing time. Tests carried out with 100 classes instead of 50 resulted in very similar results (data not shown).

We learn amino acid functional categories using the Pfam database of amino acid sequence alignments of protein domains [24]. We start by learning the amino acid stationary distribution of each rate category. We then use these distributions to classify the Pfam alignment columns, and use this classification to estimate the amino acid and codon substitution rate matrices for each class.

The stationary amino acid distributions $\pi_1, \pi_2, ..., \pi_{50}$ and class prior probability distribution τ are estimated using an unsupervised Expectation-Maximization algorithm (see Methods) in order to fit the Pfam alignments as closely as possible. Figure 2 (top) shows the amino acid distribution obtained for each of the 50 classes (see also Supplementary data). We observe that most of the expected functional classes are present among our 50

classes. First, for each amino acid, there is at least one category where that amino acid has a probability at least 0.5. These correspond to positions that tolerate little variation outside of that particular amino acid. Less constrained categories include several combinations of residues with similar properties: hydrophobic residues (ILV), small neutral residues (AST), aromatic residues (YF), positively charged residues (KR), etc. The class of negatively charged amino acids (DE) is surprisingly not directly represented, although these two amino acids show up together in several more weakly defined classes. Various categories correspond to positions under little selective pressure. Many of our classes are similar to those reported by Sjolander et al. [25] using a related procedure.

Amino acid and codon substitution rate matrices

For each of the 50 classes above, an amino acid substitution rate matrix and a codon substitution rate matrix are derived. We first compute the probability of each Pfam alignment column to belong to each of the classes, and use these to estimate the probability of amino acid and codon transitions between human and mouse sequences. Rate matrices are then derived from these empirically estimated transition probabilities matrices. The detailed procedure is described in Methods.

Figure 2 (bottom) shows two of the codon rate matrices obtained. Note how mutations toward codons encoding favored amino acids occur at a high rate, whereas substitutions away from those are rare. Notice that these rate matrices automatically take into account codon biases, as they are built from mRNA sequences. We make the assumption that the codon biases of human and mouse, which are reflected in our rate matrices, are representative of the codon biases in the other species used in our analysis. This assumption appears to hold quite well within mammals and birds, and to a lesser extent within vertebrates in general [26]. However, we recognize the fact that some genes (e.g. those requiring a high rate of translation) may have codon biases that are stronger than the average, resulting in an unexpected degree of conservation. Still, since this type of selective pressure would most likely apply to the entire transcript, it would be easily detectable using our approach, and would not result in false identification of other types of non-coding elements.

Distribution of conservation scores

We now return to the problem of identifying regions under non-coding selective pressure in a multiple alignment of orthologous coding mRNA sequences $\mathcal{X} = X_1 \ldots X_m$, where X_i is the triplet of alignment columns corresponding to the i-th codon in the alignment. Let $X_i(j)$ be the codon observed in species $j \in \{1 \ldots s\}$, where s is the number of column triplets in the alignment, and let $X_{i,p}(j)$ be the nucleotide at position p (p = 1, 2 or 3) in that codon. Given a column of orthologous codons X_i, we want to assess whether the conservation observed at position p of the codon is unexpected. To this end, we compute the posterior p-value of the observed conservation score (entropy or parsimony score) of that codon position, under the null hypothesis that the columns are only under coding selective pressure.

To describe more formally our null model of sequence evolution, we need to introduce some notation. Let $T = (V, E)$ be a binary phylogenetic tree with vertices V, edges E, root r, and with leaves numbered 1,2,...,n. Let $\lambda(u, v)$ be the length of the branch going from node u to node v, let $a(c)$ be the amino acid encoded by codon c, and let $\mathbf{Q}$ be some codon substitution rate matrix. The codon transition probability matrix for a branch (u,v) is given by $\mathbf{P}_{(u,v)} = e^{\lambda(u,v)\mathbf{Q}}$ [27]. Let b(c) be the background probability of codon c, which is assumed to be the stationary distribution of $\mathbf{Q}$. These three parameters (T, λ, $\mathbf{Q}$) describe a process that generates random but related codons at the leaves of the tree T, by drawing a codon from the stationary distribution of $\mathbf{Q}$ at the root of T and letting it mutate along the branches of the tree using the appropriate transition probability matrices. Let $C(u)$ be the random variable representing the codon that has been generated by this process at node u of the tree.

We are interested in computing the distribution of the conservation score of a given position p of a set of random orthologous codons generated at the leaves of the tree. We start by showing how to compute this distribution for the entropy score $entropy(C_p(1),C_p(2), \ldots, C_p(n))$, and later show the modifications required to do the same for the parsimony score. For a fixed codon position $p \in \{1,2,3\}$, and for any node $u \in V$, let $Y_u = (Y_u(\mathrm{A}), Y_u(\mathrm{C}), Y_u(\mathrm{G}), Y_u(\mathrm{T}))$ be a random multivariable where $Y_u(\alpha)$ is the number of nucleotides of type a at position p of the codons at the leaves of the subtree rooted at u. Notice that Y_u is only a function of the codons at the leaves of subtree(u), and not of those at the internal nodes of subtree(u). The p-value of a certain entropy score e for position p is obtained by summing the probabilities of all values of Y_r that yield an entropy score e or better:

$$\begin{aligned}\Pr[entropy(C_p(1), C_p(2), \ldots, C_p(n)) \le e] &= \sum_{\substack{\gamma_a,\gamma_c,\gamma_g,\gamma_t \in \mathbb{N}\,\text{s.t.} \\ \gamma_a+\gamma_c+\gamma_g+\gamma_t=n \\ entropy(\gamma_a,\gamma_c,\gamma_g,\gamma_t)\le e}} \Pr[Y_r = (\gamma_a,\gamma_c,\gamma_g,\gamma_t)] \\ &= \sum_{\kappa \in \mathrm{Codons}} \sum_{\substack{\gamma_a,\gamma_c,\gamma_g,\gamma_t \in \mathbb{N}\,\text{s.t.} \\ \gamma_a+\gamma_c+\gamma_g+\gamma_t=n \\ entropy(\gamma_a,\gamma_c,\gamma_g,\gamma_t)\le e}} \Pr[Y_r = (\gamma_a,\gamma_c,\gamma_g,\gamma_t) \mid C(r) = \kappa] \cdot b(\kappa) \qquad (1)\end{aligned}$$

We will show how to compute $\Pr[Y_u = (\gamma_a, \gamma_c, \gamma_g, \gamma_t) | C(u) = k]$, for every node u and all choices of γ_a, γ_c, γ_g, γ_t and k, using a dynamic programming algorithm visiting the nodes of T in post-order. When u is a leaf, we have

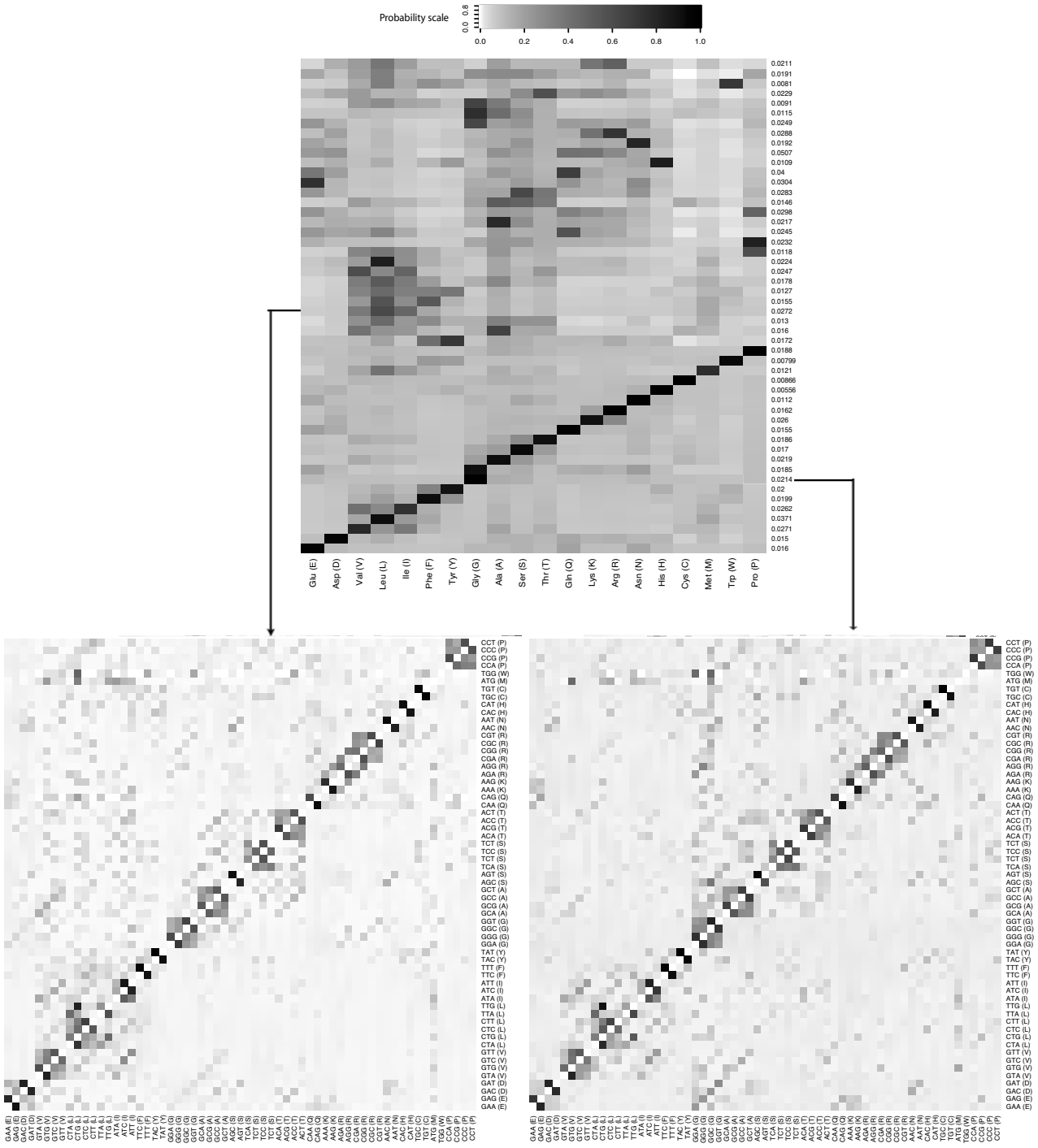

Figure 2
Mixture of codon substitution models based on amino acid functionality classes. (Top) Stationary distributions estimated from the Pfam database for the 50 functional classes. Each row corresponds to one class. The numbers on the right are the prior probability of each class. (Bottom) Two examples of codon rates matrices, where dark cells correspond to high substitution rates and light cells to low rates. The left matrix corresponds to a functional class that favors hydrophobic residues (ILV), while the right matrix comes from a class that favors the glycine amino acid (G).

$$\Pr[Y_u = (\gamma_a, \gamma_c, \gamma_g, \gamma_t) \mid C(u) = \kappa] = \begin{cases} 1 & \text{if } \gamma_a = 1, \gamma_c = \gamma_g = \gamma_t = 0 \text{ and } \kappa(p) = \text{A} \\ 1 & \text{if } \gamma_c = 1, \gamma_a = \gamma_g = \gamma_t = 0 \text{ and } \kappa(p) = \text{C} \\ 1 & \text{if } \gamma_g = 1, \gamma_a = \gamma_c = \gamma_t = 0 \text{ and } \kappa(p) = \text{G} \\ 1 & \text{if } \gamma_t = 1, \gamma_a = \gamma_c = \gamma_g = 0 \text{ and } \kappa(p) = \text{T} \\ 0 & \text{otherwise} \end{cases}$$

Define $(\gamma_a, \gamma_c, \gamma_g, \gamma_t) \oplus (z_a, z_c, z_g, z_t) := (\gamma_a + z_a, \gamma_c + z_c, \gamma_g + z_g, \gamma_t + z_t)$. Now, let u be an internal node with children v and w. Notice that $Y_u = Y_v \oplus Y_w$. We compute the desired conditional probabilities at node u based on those at nodes v and w:

$$\Pr[Y_u = (\gamma_a, \gamma_c, \gamma_g, \gamma_t) \mid C(u) = \kappa_u] = \sum_{\substack{\kappa_v, \kappa_w \in \text{Codons} \\ \Delta_v, \Delta_w \in \mathbb{N}^4 \text{ s.t.} \\ \Delta_v \oplus \Delta_w = (\gamma_a, \gamma_c, \gamma_g, \gamma_t)}} \Pr[Y_v = \Delta_v \mid C(v) = \kappa_v] \cdot \Pr[Y_w = \Delta_w \mid C(w) = \kappa_w] \cdot \mathbf{P}_{(u,v)}(\kappa_u, \kappa_v) \cdot \mathbf{P}_{(u,w)}(\kappa_u, \kappa_w) \quad (2)$$

Implementation optimizations and computational complexity analysis are given in Methods.

Distribution of parsimony scores

The method described in the previous section can be surprisingly easily modified to compute the conditional p-value of parsimony scores instead of that of the entropy score. We need to redefine the random variable $Y_u = (\gamma_a, \gamma_c, \gamma_g, \gamma_t)$ so that γ_α is now the parsimony score obtained for the nucleotides at position p of codons at the leaves of the subtree rooted at u, assuming that the nucleotide at the ancestral node u is required to be a. Note that this set of four numbers per node is exactly that computed by the Sankoff algorithm for computing parsimony scores [22]. We also redefine the $\oplus$ operator as

$$(\gamma_a, \gamma_c, \gamma_g, \gamma_t) \oplus (z_a, z_c, z_g, z_t) = (\min(\gamma_a + z_a, \gamma_a + z_{\bar{a}} + 1, \gamma_{\bar{a}} + z_a + 1, \gamma_{\bar{a}} + z_{\bar{a}} + 2),$$

$$\min(\gamma_c + z_c, \gamma_c + z_{\bar{c}} + 1, \gamma_{\bar{c}} + z_c + 1, \gamma_{\bar{c}} + z_{\bar{c}} + 2),$$

$$\min(\gamma_g + z_g, \gamma_g + z_{\bar{g}} + 1, \gamma_{\bar{g}} + z_g + 1, \gamma_{\bar{g}} + z_{\bar{g}} + 2),$$

$$\min(\gamma_t + z_t, \gamma_t + z_{\bar{t}} + 1, \gamma_{\bar{t}} + z_t + 1, \gamma_{\bar{t}} + z_{\bar{t}} + 2))$$

where $\gamma_{\bar{i}} = min_{j \neq i} \gamma_j$. Notice that this is again in direct analogy to Sankoff's algorithm. Again, $Y_u = Y_v \oplus Y_w$ and we get $parsimony(C_p(1), (C_p(2), ..., C_p(n)) = min(Y_r)$. With these redefinitions, Equation 2 holds without any modifications needed. We get

$$\Pr[parsimony(C_p(1), C_p(2), ..., C_p(n)) \leq \psi] = \sum_{\substack{\kappa \in \text{Codons} \\ \gamma_a, \gamma_c, \gamma_g, \gamma_t \in \mathbb{N} \text{ s.t.} \\ \min(\gamma_a, \gamma_c, \gamma_g, \gamma_t) \leq \psi}} \Pr[Y_r = (\gamma_a, \gamma_c, \gamma_g, \gamma_t) \mid C(r) = \kappa] \cdot b(\kappa)$$

Posterior distributions of conservation scores

Having shown how to compute the p-value of a given entropy or parsimony score under a fixed codon rate matrix, it is simple to compute posterior p-values for the case where the functional class is not known in advance. Consider a given set of aligned codons $X_i = (X_i(1), ..., X_i(n))$ encoding the set of amino acids $A_i = (A_i(1), ..., A_i(n))$. Define the unobserved variables $Z_{i,k} = 1$ if the site i belongs to functional class k, and zero otherwise. We first compute the posterior probability of each $Z_{i,k'}$ given the observed amino acids at site i:

$$\Pr[Z_{i,k} = 1 \mid A_i(1), ..., A_i(n), \mathbf{Q}_k^a] = \frac{\tau(k) \Pr[A_i(1), ..., A_i(n) \mid Z_{i,k} = 1, \mathbf{Q}_k^a]}{\sum_{k' \in \{1...50\}} \tau(k') \Pr[A_i(1), ..., A_i(n) \mid Z_{i,k'} = 1, \mathbf{Q}_k^a]},$$

where $\Pr[A_i(1), ..., A_i(n) | Z_{i,k} = 1, \mathbf{Q}_k^a]$ is computed using Felsenstein's algorithm [28], with rate matrix $\mathbf{Q}_k^a$. Finally, we obtain the posterior estimate $pv_{post}(i,p)$ for the p-value of the entropy score e observed at position p of codon i, by summing over all classes:

$$\begin{aligned} pv_{post}(i,p) &= \Pr[entropy(C_p(1), ..., C_p(n)) \leq e \mid A_i(1), ..., A_i(n)] \\ &= \sum_{k=1...50} \Pr[entropy(C_p(1), C_p(2), ..., C_p(n)) \leq e \mid \mathbf{Q}_k] \cdot \Pr[Z_{i,k} = 1 \mid A_i(1), ..., A_i(n)], \end{aligned} \quad (3)$$

and similarly for parsimony scores p-values.

Conditional p-values

An alternative to trying to guess the type of selective pressure under which a given codon evolves is to use a single codon rate matrix but subject to the constraint that the random codon generated at each leaf has to encode the amino acid that was actually observed at that leaf. This approach was originally proposed by Blanchette [18]. This model does not rely on amino acid classifications and in fact allows sites to change function during their evolution. By conditioning on the observed amino acids at the leaves of the tree, we ask: given that in species j, the codon *had* to encode amino acid $a(X_j(i))$, for each leaf $j \in \{1, ..., n\}$, is the conservation observed in X_i surprising? Notice how, compared to the mixture model approach, this model transfers the responsibility of sequence conservation even more onto the shoulders of coding selection. See Figure 3 for an example. Mathematical details are provided in Methods.

A sliding window approach

Until now, we have shown two ways to compute p-values for individual alignment columns. Since most non-coding functional elements are expected to span several consecutive positions (5-15nt for transcription factor binding sites and exonic splicing enhancers, and up to a few hundred nucleotides for RNA secondary structure elements), we can improve the sensitivity of the method by using a sim-

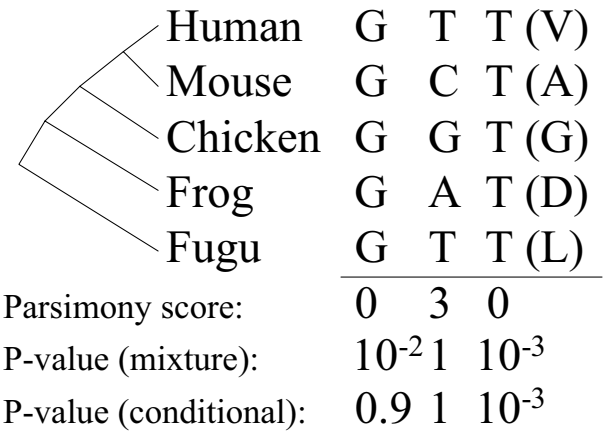

Figure 3
Example of codons where the posterior p-values under the mixture of codon models differs significantly from p-values obtained from the conditional p-value method. Amino acids V,A,G,D,L do not have any common properties, so, under the codon mixture model, the column is assigned to a functional class with little selective pressure, explaining why the p-value produced for the first codon position is small. In contrast, under the conditional p-value approach, the amino acids almost completely determine the first codon position (the only exception being the Fugu codon, encoding a leucine, which could have used a T at the first position), so a poor p-value is reported.

ple sliding window approach. For each position i, let $pv(i)$ be the p-value obtained for the i-th column of the alignment (not the i-th codon), and let w be the width of the sliding window (assume w is odd, for simplicity). We compute $S_i = \prod_{j = i-$ <Fences>Qw/2<Fences>N$...i + $ <Fences>Qw/2<Fences>N $pv(j)$. If we assume that, under the null model, $pv(j)$ is approximately uniformly distributed, a compounded p-value can be assigned to S_i: $cpv(i)$ = Pr[Product of w i.i.d. uniform (0,l) $\leq S_i$] = $S_i(1 + \sum_{j = 1...w-1} - \ln(S_i)^j/j!)$. This is the type of p-value being reported in the Applications section. It should however be noted that although the uniformity assumption holds quite well for the third codon positions, the first two codon positions are often completely determined by the codon they encode, so the range of possible p-values they can take is quite small. This results in the compounded p-values being quite conservative.

Implementation

The algorithms were implemented in C++ and the program is available upon request. A number of optimizations described in [18] have also been implemented and make the program relatively fast. In particular, a caching mechanism allows to re-use the results of computations done on previous columns. This allows the program to handle very long sequences quickly. All analyses reported here have been obtained in less two hours of computation on a desktop machine.

Analysis of simulated data

We first verify that the p-values computed by our approach have the basic properties we would expect of them. To start, we confirm that sequences evolving under the null model obtain p-values that are approximately uniformly distributed. This would be a trivial statement if the functional category of each site was known, but in the absence of such prior knowledge, the uniformity of p-values under the null model is less obvious. To this end, we simulated the evolution of a 50 kb region of DNA, with each codon belonging to one of the 50 rate categories described above. Sequences were evolved along the branches of the 69-leaf phylogenetic tree derived from the GH1 data set described below. Figure 4 shows the distribution of posterior p-values obtained at each of the three codon positions. As desired, the distribution is nearly uniform. However, we observe a depletion of columns with low p-values (<0.1), in particular for codon positions 1 and 2. This is due to the fact that, at these positions, a column that is perfectly conserved is not particularly surprising, and obtains a p-value around 0.2–0.3. To obtain a p-value distribution that is closer to uniform, one would need to use a tree with much larger total branch length. Second, we study the power of our method to detect selection on sets of aligned codons that are perfectly conserved. As expected, the power of our method to detect CRUNCS depend on the amino acid encoded by the codon. Perfectly conserved codons encoding amino acid W cannot be detected by our approach. On the other hand, codon conservation is easiest to detect among four-fold and six-fold degenerate amino acids. Among those, codons encoding amino acids P and G, both of which have chemical properties making them more difficult to exchange for other amino acids, obtain higher p-values, because their conservation can be explained to a larger extent by the amino acid encoded.

Analysis biological data

We illustrate our approach on two sets of orthologous mRNA sequences: a set of 69 vertebrate growth hormone 1 (*GH1*) sequences, and a set of 13 vertebrate *CORTBP2* (also known as *CTTNBP2*) sequences (see Supplementary data).

GH1 was one of the first gene shown to harbor an exonic splicing enhancer, in cow [29]. The availability of a large number of orthologous sequences makes it ideal for our study. Figure 5 shows the compounded p-values obtained from the mixture-based method, for the parsimony score, using a sliding window with w = 9. The human region orthologous to a known exonic splicing enhancer in the

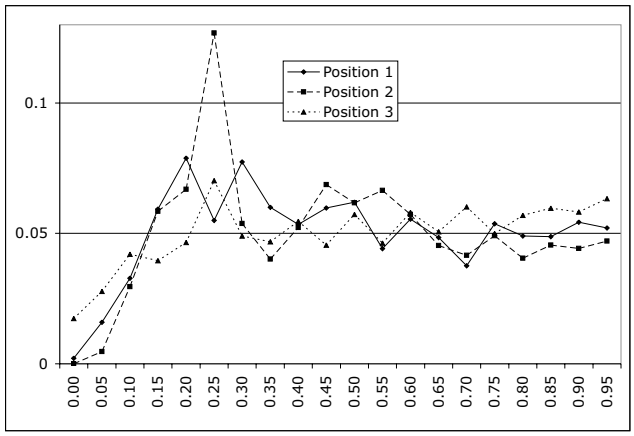

Figure 4
Distribution of the p-values obtained on a set of 50-kb sequences evolving according to the null model. The tree used for the simulation is the same as that for the GH1 dataset (see below).

cow, located around position 577 [29], is clearly identified, obtaining a compounded p-value of about 2×10^{-3}. Although this region could have been identified on the basis of parsimony scores alone (gray curve on Figure 5), it is highlighted much more clearly by the posterior p-values.

Figure 6 compares the posterior p-values obtained for the entropy and parsimony scores, under the mixture model of codon evolution, on the growth-hormone 1 data set. The correlation between the two is quite obvious, especially for very small p-values. This is in part due to the fact that alignment columns that are perfectly conserved obtain the same p-values from the two methods. On the other hand, many more sites obtain p-values between 0.01 and 0.1 using the parsimony score than using the entropy score. These are sites that have undergone a small number of substitutions early in vertebrate evolution, and that obtain a good parsimony score but a poor entropy score, as in Figure 1. In this data set, there are no sites that obtain a good entropy p-value and a poor parsimony p-value. Since the parsimony score p-values can be computed much faster and since they appear to be strictly more sensitive than entropy p-values, they seem to be the method of choice in most situations.

Figure 7 compares the p-values obtained for the parsimony score under the mixture model and to those obtained under the conditional p-value approach. Although the two are clearly correlated, there are some important differences. First, most of the p-values obtained for the 1st and 2nd codon positions under the conditional probability model are very close to 1, because these two codon positions are often completely determined by the amino acids observed at the leaves. Under the mixture model, the p-values obtained at these positions will depend on the variability of the amino acids observed, and in general will be smaller. This is also true, to a lesser extent, for the third codon position.

Finally, the analysis of the *CORTBP2* transcript is particularly intriguing. Little is known about the post-transcriptional regulation of this gene. We find that its mRNA contains a very large region (roughly between positions 1500 and 2200, see Figure 8 and Supplementary material), which obtain fairly low p-values. This region is much too large to be a binding site, and we conjecture that it may form some large RNA secondary structure affecting the pre-mRNA splicing or the mRNA stability, localization or translation. Notice that in this case, a simple parsimony score is not sufficient for identifying the region, as many other regions of the transcript are well equally well conserved.

Conclusion

With the many genome sequencing projects rapidly producing vertebrate genomic data, comparative genomic approaches are becoming increasingly powerful. In the case of CRUNCS, additional data is often available in the form of ESTs and cDNAs. We believe that within one year or two, there will be sufficient data for accurate detection of CRUNCS in vertebrate genes, using methods like those described here. Once many CRUNCS will have been detected, the next step will of course be to assign functions to these elements. Although the last word will remain with experimentalists, we have good hopes that more advanced bioinformatics approaches will yield insights into these questions.

Finally, we expect that organisms that are under severe genome size constraints, in particular bacteria and viruses, will more often use CRUNCS. We believe that our approaches will prove particularly fruitful to analyze these genomes.

Methods

Estimating amino acid stationary distributions for each class

The Pfam database consists of a set of multiple alignments of homologous protein domain sequences. For some domains, the database contains several sequences that are very closely related. To reduce biases due to this over-representation, one member of any pair of domain sequences that share more than 60% identity is discarded. Let D_1, ..., D_m be the set of alignment columns in this reduced Pfam database, let S_i be the number of species in alignment column i, let $D_i(j)$ be the amino acid from species j in column i, and let $E_i(j)$ be the codon encoding that amino acid in

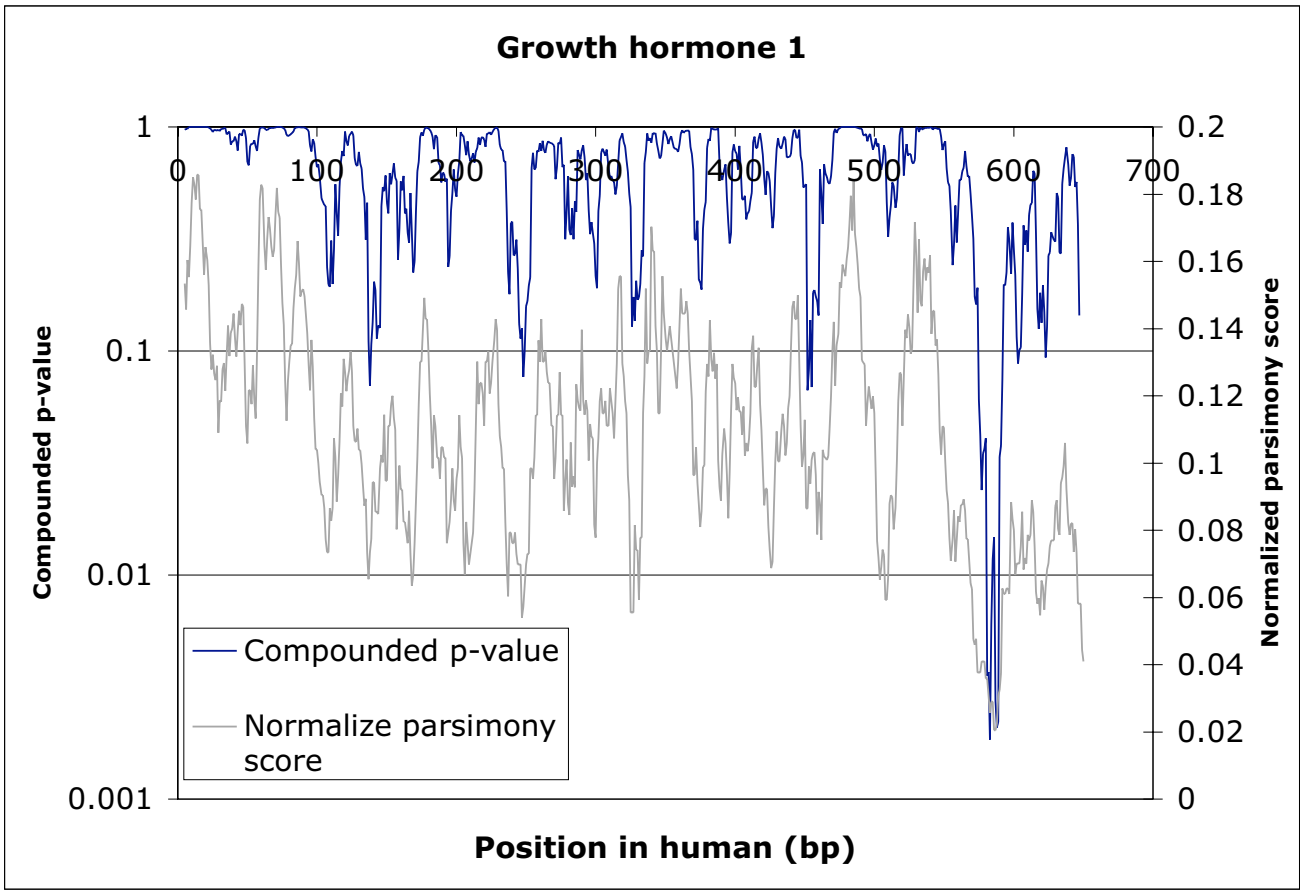

Figure 5
Dark curve: compounded parsimony p-values for the *GH1* gene under the mixture model, using a window size of $w = 9$ nucleotides. Light curve: average parsimony score per alignment column, normalized by the number of leaves in the tree (which varies from site to site, due to gaps in the alignment).

the mRNA of species j. Starting from Pfam 17.0, we have $m = 2292182$ amino acid alignment columns.

For simplicity, we assume that the amino acids in a given column are drawn independently from the same unknown amino acid distribution. Let $\pi_1, \pi_2, \ldots, \pi_{50}$ be a set of amino acid distributions, where $\pi_k(\alpha)$ is the probability of amino acid a in class k. Let $\tau(k)$ be the prior probability of class k. The class membership of column i is unknown. Let $Z_i(k)$ be a hidden variable that takes value 1 if column i belongs to class k, and 0 otherwise. We have $\Pr[D_i|Z_i(k) = 1] = \prod_{j=1\ldots s_i} \pi_k(D_i(j))$.

We search for the distributions $\pi_1, \ldots, \pi_{50}$ and prior probabilities τ that maximize $\Pr[D_1 \ldots D_m] = \prod_{i=1\ldots m} \Pr[D_i] = \prod_{i=1\ldots m} \sum_{k=1..50} \Pr[D_i|(k) = 1]\tau(k)$. This is achieved using a simple EM algorithm for learning mixtures of multinomial distributions, using the following update formulas to go from iteration t to iteration $t + 1$.

$$\pi_k^{(t+1)}(\alpha) \leftarrow (1/\zeta) \sum_{i=1\ldots m} \frac{\prod_{j=1\ldots s_i} \pi_k^{(t)} \cdot D_i(j) \cdot \tau^{(t)}(k)}{\sum_{k'=1\ldots 50} \prod_{j=1\ldots s_i} \pi_{k'}^{(t)} \cdot D_i(j) \cdot \tau^{(t)}(k')} N_i(\alpha)$$

and

$$\tau^{(t+1)}(k) \leftarrow (1/\zeta') \sum_{i=1\ldots m} \frac{\prod_{j=1\ldots s_i} \pi_k^{(t)} \cdot D_i(j) \cdot \tau^{(t)}(k)}{\sum_{k'=1\ldots 50} \prod_{j=1\ldots s_i} \pi_{k'}^{(t)} \cdot D_i(j) \cdot \tau^{(t)}(k')},$$

where $N_i(\alpha)$ is the number of occurrences of amino acid α in column D_i and where ζ and ζ' are normalizing constants that ensure that the probabilities sum to one.

Initialized with noisy uniform distributions, the algorithm converges quickly (less than 50 iterations) to the

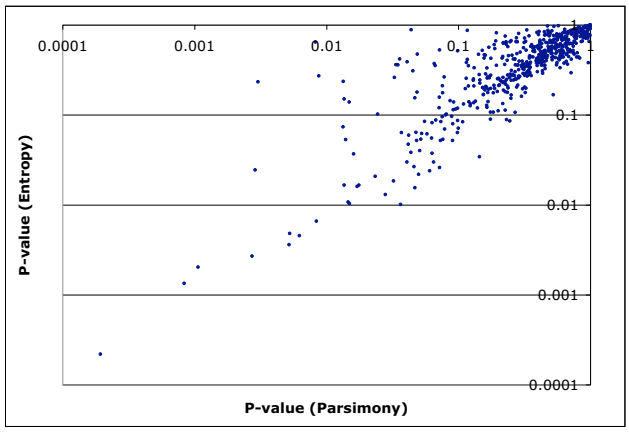

Figure 6
Comparison of the posterior p-values obtained from the entropy and parsimony scores, under the mixture of codon models, on the *GH1* gene.

local optimum depicted in Figure 2. Ten restarts from other random initial conditions converged to very similar distributions, so the first one was retained.

Estimating amino acid and codon rate matrices

Once the amino acid distributions $\pi_1, \ldots, \pi_{50}$ and prior distribution τ are learned, we can use them to compute the posterior probability of each class k for each alignment column D_i:

$$\Pr[Z_i(k) = 1 \mid D_i] = (\tau(k)/\zeta'') \prod_{j=1\ldots s} \pi_k(D_i(j)),$$

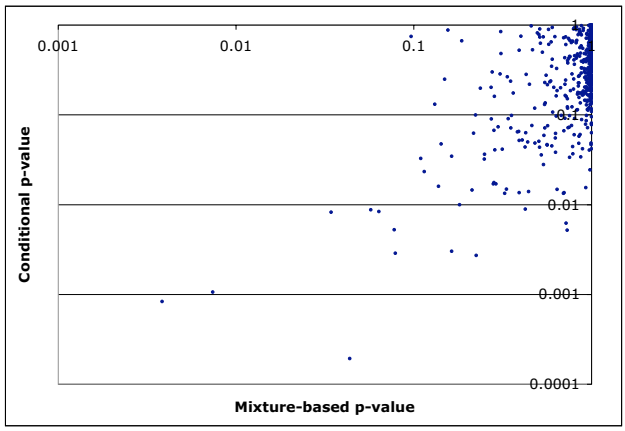

Figure 7
Comparison of the p-values obtained from the parsimony scores, under the mixture of codon models and under the conditional probability computation.

where ζ'' is another normalizing constant. For simplicity, the amino acid and codon substitution rate matrices $\mathbf{P}_k^a$ and $\mathbf{P}_k^c$ are estimated for each class k based only on the observed substitutions between human and mouse sequences (however, the class posterior probabilities of each column are based on all available species). Any column that does not include an entry for these two species is excluded. Let $E_i(human)$ and $E_i(mouse)$ be the codons encoding amino acid $D_i(human)$ and $D_i(mouse)$ in the corresponding mRNA sequences. The transition matrices are then estimated as follows:

$$\mathbf{P}_k^a(\alpha,\beta) = \frac{\sum_{i:D_i(human)=\alpha, D_i(mouse)=\beta} \Pr[Z_i(k)=1 \mid D_i]}{\sum_{i:D_i(human)=\alpha} \Pr[Z_i(k)=1 \mid D_i]}$$

$$\mathbf{P}_k^c(\alpha,\beta) = \frac{\sum_{i:E_i(human)=\alpha, E_i(mouse)=\beta} \Pr[Z_i(k)=1 \mid D_i]}{\sum_{i:E_i(human)=\alpha} \Pr[Z_i(k)=1 \mid D_i]}$$

Finally, we estimate the instantaneous rate matrix $\mathbf{Q}_k^a(\alpha, \beta) = \ln \mathbf{P}_k^a(\alpha, \beta)/t_{(h,m)}$, where $t_{(h,m)}$ is the expected number of substitutions per site between human and mouse, in neutrally evolving DNA. Codon rate matrices are obtained in a similar manner.

Note that the codon substitution rates we obtain are slightly underestimating the true rates for sites evolving only under coding selective pressure, because some sequences in Pfam are likely to be evolving slower due to non-coding selective pressure. However, we expect that this underestimation is negligible as the fraction of such sites is likely to be small. In any case, underestimating the rates will only cause conservative estimates of the conservation p-values.

Computing entropy and parsimony score p-values

The probabilities $\Pr[Y_u = (\gamma_a, \gamma_c, \gamma_g, \gamma_t) \mid C(u) = k_u]$ are stored in a hash table associated to each node, indexed by the quintuplet $(\gamma_a, \gamma_c, \gamma_g, \gamma_t, k_u)$. Only non-zero probabilities are stored. To compute these probabilities for a node u with children v and w, it is simpler to enumerate all pairs $(\gamma_a, \gamma_c, \gamma_g, \gamma_t, k_v)$, $(z_a, z_c, z_g, z_t, k_w)$ of quintuplets from the hash tables of the two children, and add the proper quantity to the entry $((\gamma_a, \gamma_c, \gamma_g, \gamma_t) \oplus (z_a, z_c, z_g, z_t), k_u)$ of the hash table at u, for all choices of k_u. Please see [18] for more details.

To study the complexity of the resulting algorithm for the entropy p-value computation, observe that the hash table

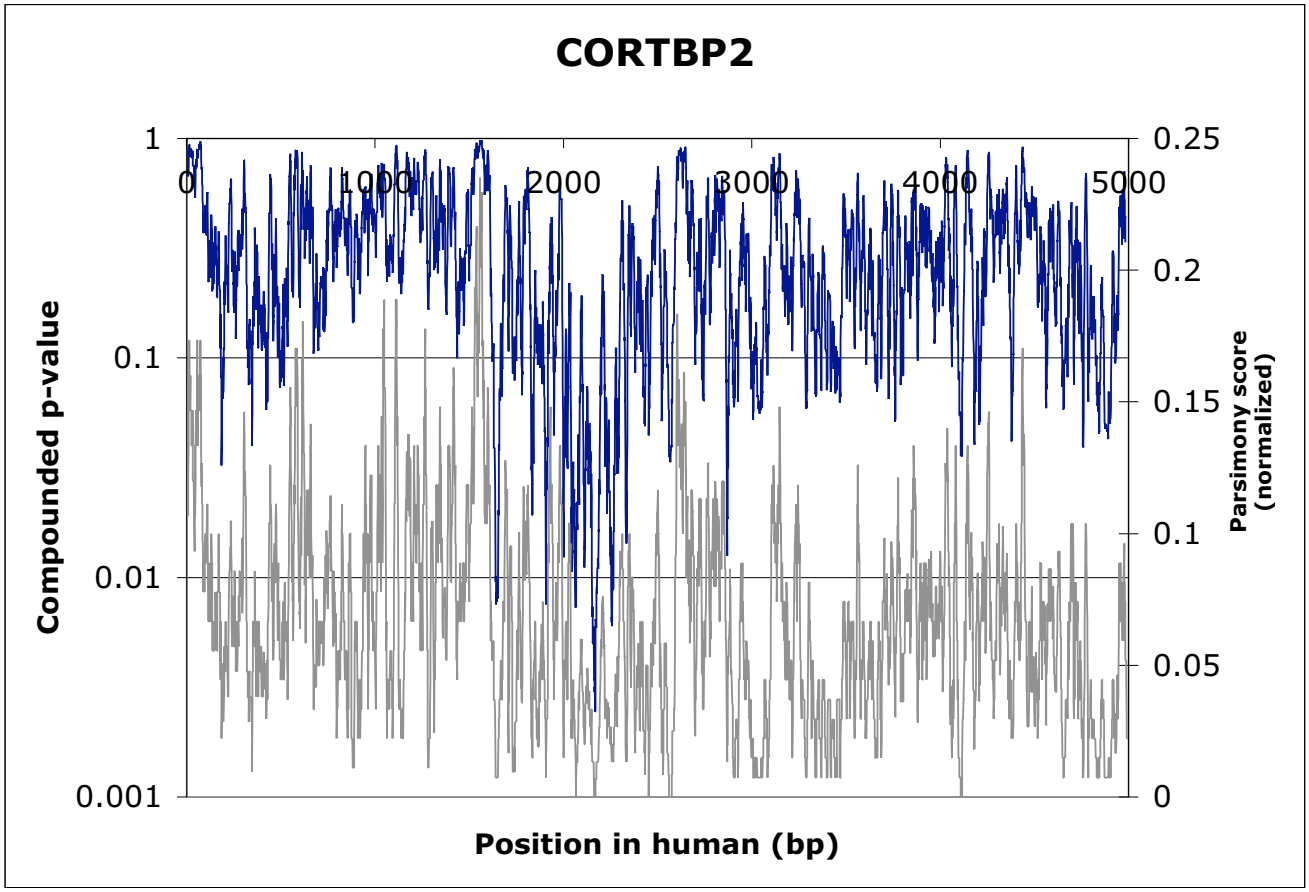

Figure 8
Dark curve: compounded parsimony p-values for the *CORTBP2* gene, using a window size of $w = 15$ nucleotides. Light curve: average parsimony score per alignment column, normalized by the number of leaves in the tree (which varies from site to sites, due to gaps in the alignment).

associated with node v whose subtree contains $l(v)$ leaves will contain at most $64 \cdot \binom{l(v)+3}{3} \in O(l(v)^3)$ non-zero entries and thus computing all entries of the hash table for node u takes $O(l(v)^3 l(w)^3)$ time. Depending on the topology of T, computing the hash tables of all nodes takes between $O(n^4)$ time for a completely skewed tree and $O(n^6)$ time for a balanced tree. The time complexity analysis applies to both the posterior p-value and the conditional p-value computations. However, for posterior p-values, this computation needs to be repeated 50 times, once for each rate matrix.

The hash table implementation works well in the case of parsimony score p-value computation too, especially with the following optimizations. First, for binary trees, the only choices of $\gamma_a, \gamma_c, \gamma_g, \gamma_t$ that may have non-zero probability are those where $|\gamma_i - \gamma_j| \leq 2$ for all $i, j \in \{A, C, G, T\}$ (this is a direct consequence of the $\oplus$ operator). When evaluating the p-value for a parsimony score ψ, choices of $\gamma_a, \gamma_c, \gamma_g, \gamma_t$ such that $min(\gamma_a, \gamma_c, \gamma_g, \gamma_t) > \psi$ are not affecting the final p-value and can be safely ignored. Therefore, the hash table associated with node v contains $O(\psi) = O(n)$ entries, and computing all entries for node u from those of nodes v and w take $O(l(u) \cdot l(v))$. Thus, one can compute all entries of all tables in $O(n^2)$, irrespective of the tree topology. More details can be found in [18].

Conditional P-values

The method of conditional p-values, introduced by Blanchette [18], can be summarized as follows. The method computes the following conditional p-value:

$$pv_{cond}(i, p) = \Pr[entropy\gamma(C_p(1), \ldots, C_p(n)) \leq entropy\gamma(X_{i'p}(1), \ldots, X_{i'p}(n)) | a(C(1)) = a(X_i(1)), \ldots, a(C(n)) = a(X_i(n))]$$

This p-value can be computed in a manner that is similar to the Equation 2. Let us denote by $l(u)$ the set of leaves in the subtree rooted at u and let us write the set of conditions for the subtree rooted at u as $A(u) = \wedge_{j \in l(u)}(a(C(j)) = a(X_i(j)))$. We get

$$pv_{cond}(i,p) = \sum_{\kappa \in \text{Codons}} \sum_{\substack{\gamma_a,\gamma_c,\gamma_g,\gamma_t \in \mathbb{N} \text{ s.t.} \\ \gamma_a+\gamma_c+\gamma_g+\gamma_t=n \\ entropy(\gamma_a,\gamma_c,\gamma_g,\gamma_t) \le entropy(X_{i,p}(1),\ldots,X_{i,p}(n))}} \Pr[Y_r = (\gamma_a,\gamma_c,\gamma_g,\gamma_t) \mid C(r) = \kappa, A(r)] \cdot \Pr[C(r) = \kappa \mid A(r)]$$

and

$$\Pr[Y_u = (\gamma_a,\gamma_c,\gamma_g,\gamma_t) \mid C(u) = \kappa_u, A(u)] = \sum_{\substack{\kappa_v,\kappa_w \in \text{Codons} \\ \Delta_v,\Delta_w \in \mathbb{N}^4 \text{ s.t.} \\ \Delta_v \oplus \Delta_w = (\gamma_a,\gamma_c,\gamma_g,\gamma_t)}} \left(\begin{array}{l} \Pr[Y_v = \Delta_v \mid C(v) = \kappa_v, A(v)] \cdot \left[\dfrac{P_{(u,v)}(\kappa_u,\kappa_v) \Pr[A(v) \mid C(v) = \kappa_v]}{\sum_{d \in \text{Codons}} P_{(u,v)}(\kappa_u,d) \Pr[A(v) \mid C(v) = d]} \right] \cdot \\ \Pr[Y_w = \Delta_w \mid C(w) = \kappa_w, A(w)] \cdot \left[\dfrac{P_{(u,w)}(\kappa_u,\kappa_w) \Pr[A(w) \mid C(w) = \kappa_w]}{\sum_{d \in \text{Codons}} P_{(u,w)}(\kappa_u,d) \Pr[A(w) \mid C(w) = d]} \right] \end{array} \right)$$

where $\Pr[C(u) = k|A(u)]$, is computed using Felsenstein's algorithm and Bayes rule. As before, a dynamic programming algorithm proceeding in a post-order traversal of the tree allows the computation of all terms required.

Authors' contributions

Chen implemented the program for the posterior p-value computation, for learning amino acid functional classes, and for learning the rate matrices associated to each class, and did some of the data analysis. Blanchette was responsible for the original ideas and mathematical derivations, for some of the data analysis, and for writing the paper.

Supplementary data

Multiple alignments, phylogenetic trees and detailed results are available at [30].

Acknowledgements

We would like to thank Martin Tompa, Saurabh Sinha, Adam Siepel, and David Haussler for useful discussions early in this project. We also thank the editor Hervé Philippe and one anonymous referee for their hard work on this manuscript and for their useful suggestions.

This article has been published as part of *BMC Evolutionary Biology* Volume 7 Supplement 1, 2007: First International Conference on Phylogenomics. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcevolbiol/7?issue=S1.

References

1. The International Mouse Genome Sequencing Consortium: **Initial sequencing and comparative analysis of the mouse genome.** *Nature* 2002, **420(6915):**520-62.
2. Margulies EH, Blanchette M, Haussler D, Green ED: **Identification and characterization of multi-species conserved sequences.** *Genome Research* 2003, **13(12):**2507-2518.
3. Blanchette M, Bataille A, Chen X, Poitras C, Laganiere J, Lefebvre C, Deblois G, Giguere V, Ferretti V, Bergeron D, Coulombe B, Robert F: **Genome-wide computation prediction of transcriptional regulatory modules reveals new insights into human gene expression.** *Genome Research* 2006, **16(5):**656-68.
4. King DC, Taylor J, Elnitski L, Chiaromonte F, Miller W, Hardison RC: **Evaluation of regulatory potential and conservation scores for detecting cis-regulatory modules in aligned mammalian genome sequences.** *Genome Res* 2005, **15(8):**1051-1060.
5. Hastings M, Krainer A: **Pre-mRNA splicing in the new millenium.** *Current Opinion in Cell Biology* 2001, **13:**302-309.
6. Fairbrother W, Yeh R, Sharp P, Burge C: **Predictive identification of exonic splicing enhancers in human genes.** *Science* 2002, **297:**1007-1013.
7. Meisner NC, Hackermuller J, Uhl V, Aszodi A, Jaritz M, Auer M: **mRNA openers and closers: modulating AU-rich element-controlled mRNA stability by a molecular switch in mRNA secondary structure.** *Chembiochem* 2004, **5(10):**1432-1447.
8. Jansen RP: **mRNA localization: message on the move.** *Nat Rev Mol Cell Biol* 2001, **2(4):**247-256.
9. Kozak M: **Determinants of translational fidelity and efficiency in vertebrate mRNAs.** *Biochimie* 1994, **76(9):**815-821.
10. Kozak M: **Regulation of translation via mRNA structure in prokaryotes and eukaryotes.** *Gene* 2005, **361:**13-37.
11. Meyer IM, Miklos I: **Statistical evidence for conserved, local secondary structure in the coding regions of eukaryotic mRNAs and pre-mRNAs.** *Nucleic Acids Res* 2005, **33(19):**6338-6348.
12. Lin C, Tam RC: **Transcriptional regulation of CD28 expression by CD28GR, a novel promoter element located in exon 1 of the CD28 gene.** *J Immunol* 2001, **166(10):**6134-6143.
13. Tagle D, Koop B, Goodman M, Slightom J, Hess D, Jones R: **Embryonic ε and γ globin genes of a prosimian primate (Galago crassicaudatus) nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints.** *Journal of Molecular Biology* 1988, **203:**439-455.
14. Blanchette M, Tompa M: **Discovery of Regulatory elements by a computational method for phylogenetic footprinting.** *Genome Research* 2002, **12:**739-748.
15. Sorek R, Ast G: **Intronic sequences flanking alternatively spliced exons are conserved between human and mouse.** *Genome Res* 2003, **13(7):**1631-1637.
16. Margulies EH, Vinson JP, Miller W, Jaffe DB, Lindblad-Toh K, Chang JL, Green ED, Lander ES, Mullikin JC, Clamp M, Program NISCCS: **An initial strategy for the systematic identification of functional elements in the human genome by low-redundancy comparative sequencing.** *Proc Natl Acad Sci USA* 2005, **102(13):**4795-4800.
17. 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 Res* 2005, **15(8):**1034-1050.
18. Blanchette M: **A comparative analysis method for detecting binding sites in coding regions.** *Proceedings of the seventh annual international conference on computational molecular biology* 2003.
19. Stojanovic N, Florea L, Riemer C, Gumucio D, Slightom J, Goodman M, Miller W, Hardison R: **Comparison of five methods for finding conserved sequences in multiple alignments of gene regulatory regions.** *Nucleic Acids Res* 1999, **27(19):**3899-910.
20. Bailey TL, Elkan C: **Unsupervised learning of multiple motifs in biopolymers using expectation maximization.** *Machine Learning* 1995, **21(1–2):**51-80.
21. Fitch WM: **Toward Defining the Course of Evolution: Minimum Change for a Specified Tree Topology.** *Systematic Zoology* 1971, **20:**406-416.
22. Sankoff DD: **Minimal mutation trees of sequences.** *SIAM Journal on Applied Mathematics* 1975, **28:**35-42.
23. Lartillot N, Philippe H: **A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process.** *Mol Biol Evol* 2004, **21(6):**1095-109.
24. Bateman A, Coin L, Durbin R, Finn RD, Hollich V, Griffiths-Jones S, Khanna A, Marshall M, Moxon S, Sonnhammer ELL, Studholme DJ, Yeats C, Eddy SR: **The Pfam protein families database.** *Nucleic Acids Res* 2004:D138-D141.
25. Sjolander K, Karplus K, Brown M, Hughey R, Krogh A, Mian IS, Haussler D: **Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology.** *Comput Appl Biosci* 1996, **12(4):**327-345.
26. Nakamura Y, Gojobori T, Ikemura T: **Codon usage tabulated from international DNA sequence databases: status for the year 2000.** *Nucleic Acids Res* 2000, **28:**292.
27. Karlin S, Taylor H: *A first course in stochastic processes* second edition. *Academic Press*; 1975.

28. Felsenstein J: **Evolutionary trees from DNA sequences: a maximum likelihood approach.** *Journal of Molecular Evolution* 1981, **17:**368-376.
29. Dirksen WP, Li X, Mayeda A, Krainer AR, Rottman FM: **Mapping the SF2/ASF binding sites in the bovine growth hormone exonic splicing enhancer.** *J Biol Chem* 2000, **275(37):**29170-29177.
30. **Supplemental material** [http://www.mcb.mcgill.ca/~blanchem/CRUNCS]