Using Multiple Alignments and Phylogenetic Trees to Detect RNA Secondary Structure

Brad Gulko
University of California at Santa Cruz, Department of Computer Engineering
Lepton Incorporated
bgulko@LeptonCorp.com

David Haussler
University of California at Santa Cruz, Department of Computer Science
haussler@cse.ucsc.edu

July - 1995

Abstract: We describe a statistical method to determine if a pair of columns in a multiple alignment of a homologous family of RNA sequences shows evidence of being base paired. The method makes explicit use of a given phylogenetic tree for the sequences in the alignment. It is tested on a multiple alignment of 16S rRNA sequences with good results.

Introduction and Overview of Methods

Most present techniques for RNA secondary structure prediction are based either on energy minimization or on comparative sequence analysis. Energy minimization methods have had less success on large RNA molecules [1 Jacobson-93] [2 Zuker-91] [3 Zuker-84] [4 Tinoco-71], so comparative sequence analysis is the method of choice here* [5 Han-93] [6 Le-91]. Until now, comparative sequence methods have either required substantial manual intervention [7 James-89] [8 Woese-83], or were more fully automated, but overlooked information about the phylogenetic relationships among the sequences in the RNA multiple

---

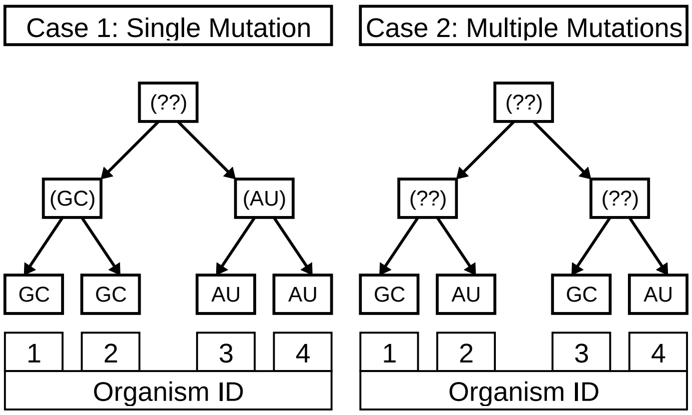
* Some hybrid methods involving both comparative sequence analysis and Energy Minimization have been attempted [5 Han-93] [6 Le-91].

alignment[†]. Among the many methods of the later type are RNA secondary structure predictors based on statistical measures of dependency such as mutual information. These methods impute base pairing between two columns of a multiple alignment when the columns are found to have a high degree of statistical dependency [9 Cary-95] [10 Klinger-93] [11 Gutell-92] [12 Waterman-89]. Some recent algorithms in this family, based on stochastic context-free grammars, also take into account information about neighboring columns in a multiple alignment [13 Lefebvre-95] [14 Eddy-94] [15 Sakakibara-94] [16 Grate-94]. In order to allow fully automated systems to make use of phylogenetic information, we develop a statistical model of the evolutionary process embodied in the phylogenetic tree. This model, which we call the *Tree Model* is then applied to pairs of columns of the multiple alignment. The Tree Model is designed to be used as a subroutine to determine if a pair of columns shows strong evidence of base-pairing in the underlying secondary structure common to the sequences in a multiple alignment. To determine the entire common secondary structure, this subroutine might be imbedded in a larger RNA structure discovery system such as [17 Grate-95] [9 Cary-95].

To clarify the setting in which the tree model is applied, suppose that we have constructed a multiple alignment of several homologous RNA sequences with each sequence on a separate row. We also have a classical phylogenetic tree (T) describing the phylogenetic relationships among these sequences. We now select two columns in this multiple alignment (a *column duo*, d). Our goal is to determine whether or not these columnar positions are *base paired.* As we are not merely examining a single molecule, the base pairing we are looking for is not the classical Watson-Crick style pairing between two individual nucleic acid molecules. Rather, we are looking for some presumed common secondary (or tertiary) pairing structure shared by all of the RNA molecules in our homologous multiple alignment. In other words, we must decide if the nucleotides in the selected multiple alignment positions interact in such a way that they form part of the common secondary structure of the family. The presence of this secondary structure is typically associated with the presence of a helix in the molecules of a homologous family of RNA.

[†] Some methods do use a heuristic sequence weighting to reduce bias in certain statistical measures.

Two possible scenarios of this are illustrated in *Figure 1*. To simplify this example, we only allow the base pairs GC and AU. Both scenarios have the same tree structure *T*, but have differing column duo data *d*. For each case, the column duo data is shown as labels in the leaves of the tree and in the *Alignment Data* on



Organism ID	Case 1	Case 2
1	GC	GC
2	GC	AU
3	AU	GC
4	AU	AU

Figure 1: Relationship Between Phylogenetic Tree and Multiple Alignment.

The above graphs show two multiple alignment column duos (4 columns) applied to the leaves of a single phylogenetic tree. The internal nodes of the tree do not correspond to organisms in the multiple alignment, rather they are unseen genetic progenitors who's genetic makeup's are inferred statistically from a those of their offspring, and a given mutation process. Each of the 4 organisms represented in the multiple alignment is denoted by a separate ID number.

the right. In both cases the data (*d*) consists of two occurrences of the nucleotide duo GC and two occurrences of the duo AU. However, in Case 1, both organisms (sequences) having GC as their contribution to *d* are related by a common parent, as are both sequences contributing AU. In Case 2, each sequence contributing GC shares a common parent with a sequence contributing AU. In both cases the presence of GC and AU duos is evidence that the duo may be base paired, but this evidence is stronger in Case 2 than it is in Case 1. This is because Case 2 requires at least two mutations of the form AU→GC or GC→AU, while Case 1 requires only one such mutation. Such combined mutations that preserve Watson-Crick base-pairing are referred to as *compensatory mutations*. Using the Tree Model, we can now quantify how much stronger the evidence for base-pairing is in Case 2 than in Case 1. The model may be applied to classify column duos as either base-paired or not base paired. We tested the model using column duos from an alignment of 1375 16S rRNA sequences obtained from the ribosomal database project [18 RDP-93] and found it to have a classification accuracy in excess of 90%. Accuracy rises to more than 99% when highly conserved column duos are removed to reduce data degeneracies. We show by direct comparison that the Tree Model method performs better than the mutual information methods. The results we obtain also compare favorably with the 60%-80% accuracies reported in previous work [19 Zuker-91] [20 Pieter-90] [21 Jaeger-90] through the use of energy minimization, manual comparative sequence analysis any their hybrids[‡].

Using the Tree Model

The evidence that a column duo d is base-paired is calculated as the log-likelihood ratio:

$$\log\left(\ P(d|Model_{pair} \wedge T)\ /\ P(d|Model_{nopair} \wedge T)\ \right)$$

The likelihood $P(d|Model_{pair})$ represents the probability that the data d would be generated at the leaves of phylogenetic tree T, assuming a particular mutation model $Model_{pair}$ that favors compensatory mutations [22 Felsenstein-81]. The likelihood $P(d|Model_{nopair})$ is similar, except that mutation model $Model_{nopair}$ does not favor compensatory mutations. $Model_{nopair}$ is constructed assuming that the mutation processes for every column duo in a multiple alignment is independent, and hence compensatory mutations are not favored. In the simple case that evolutionary times are the same on all branches of the phylogenetic tree, a mutation model is comprised of two components, a 16 by 16 matrix ρ and a 16 element vector φ. The matrix ρ provides the probability that any of the 16 possible nucleotide duos will become another of the 16 possible nucleotide duos over the span of time represented by one phylogenetic tree branch. The diagonal elements of this matrix represent the probabilities that a given nucleotide duo will not mutate (i.e. AU$\rightarrow$AU) while the off-diagonal elements hold the probabilities that a nucleotide duo will mutate (i.e. AU$\rightarrow$GC). The vector φ contains the prior probabilities of observing a given nucleotide duo in a node of the phylogenetic tree. An evolutionary reconstruction consists of the determination of a probability distribution over each possible ancestral nucleotide duo in the tree T. We assume that the mutations from a given node of the phylogenetic tree are independent, given that node's nucleotide composition. We also define an *evolutionary reconstruction* (R) of d to be an initial nucleotide duo for the root node of the phylogenetic tree coupled with a set of mutations leading from the root node to all of the observed nucleotide duos (d) at the leaves of T. Thus, the probability P($d \wedge$

[‡] This latter comparison is not completely equitable as the prior distributions of Paired versus NoPaired column duos my vary between methods. For more information on the effects of this asymmetry see the *Comparison of Methods* section of this paper, as well as Appendix B of [30 Gulko-95].

$R|Model_{pair} \wedge T$) of a specific evolutionary reconstruction for column duo d is just the prior probability of finding the specified nucleotide duo of the root ancestral node (from φ) multiplied by the product of the probabilities of each mutation in the reconstruction (from ρ). The probability that the observed data d is generated at the leaves of the tree, $P(d|Model_{pair} \wedge T)$, can be calculated by summing the probabilities of all possible evolutionary reconstructions for d,

$$P\left(d \mid Model_{pair} \wedge T\right) = \sum_{R} P\left(d \wedge R \mid Model_{pair} \wedge T\right).$$

As the number of possible evolutionary reconstructions grows exponentially with the size of T, this calculation is not directly feasible. However, by exploiting independence on the branches, this calculation can be done much more efficiently by dynamic programming [22 Felsenstein-81]. The basis for this process is explained below.

In the case that not all branch lengths in the evolutionary tree are the same, it is reasonable to use different powers of a mutation matrix (ρ) to describe the mutation process on each branch. Given a mutation rate matrix ρ(1) for a unitary span of time, the mutation matrix for a phylogenetic tree branch of length t could be calculated as $\rho(t) = \rho(1)^t$. However, it appears difficult to construct an adequate learning model for ρ(1) from a set of training data and a phylogenetic tree with varying branch length. As the Tree Model must automatically calculate its parameters from a set of training data, we use a discrete approximation for t in the continuous function $\rho(t)$. This approximation is accomplished by binning the possible branch lengths (t) into 6 discrete ranges, and using a single 16 by 16 mutation matrix for all branch lengths within a single discrete range. Thus, the $Model_{pair}$ actually includes six 16 by 16 matrices, rather than just one. In the remaining description of the method, we will not dwell on this technicality. Rather, we consider only the simple case that all branch lengths in the tree are the same.

One key issue is how to determine the parameters of the 16 by 16 matrix ρ and the vector of 16 prior probabilities φ for the model $Model_{pair}$. To accomplish this, we have used maximum likelihood estimation. We collected 472 column duos that were labeled as base-paired in the multiple alignment obtained from RDP[§] [23 Macke-93]. This set of column duos (D) was filtered and split into disjoint sets for cross validation purposes D_{train} and D_{test}. The column duos in D_{train} were used as a training set to estimate the parameters of $Model_{pair}$. Specifically, we

---

§ Actually, these were labeled as base-paired for the *E. coli* sequence in this alignment. Thus, some of the recorded column duos might not actually be base-pairing positions for sequences which differ substantially in structure from *E. coli* 16S rRNA.

calculated the parameters of $Model_{pair}$ so as to maximize the joint likelihood of the training data

$$\prod_{d \in D_{train}} \mathrm{P}(d \mid Model_{pair} \wedge T)\,.$$

While conceptually simple, this process is technically complex. This complexity stems from the fact that we do not know the explicit evolutionary histories for the training sequences. If we did, we could merely count the number of times each of the possible mutations occurred, and then set the parameters of the mutation matrix accordingly. To circumvent this lack of information, we apply the general statistical method of Expectation Maximization (*EM*). This method allows us to calculate expectation values for the desired parameters in such cases where there are critical unobserved (or *latent*) variables in the likelihood formula [24 Dempster-77]. In EM, initial parameter values are assumed. This initial estimate is then employed to collect sufficient statistics for the latent variables. Finally, the target parameter values are updated (or *reestimated*) to maximize the likelihood of the data given the observed statistics. This process is repeated until a local optimum of the likelihood function is reached. While EM is guaranteed to converge to a locally maximal likelihood, it is not guaranteed to find a global optimum. In our case, the sufficient statistics are the expected number of times each mutation occurs, calculated by considering (implicitly) all possible evolutionary reconstructions for each column duo $d \in D_{train}$. Again, this can only be done efficiently using dynamic programming methods. These dynamic programming methods are similar to, but somewhat more complex than the dynamic programming methods used to calculate the likelihood $\mathrm{P}(d \mid Model_{pair} \wedge T)$. The mutation frequency calculation is analogous to the inside-outside calculations done to estimate the parameters of a stochastic context-free grammar [25 Lari-90], which is in turn a generalization of the forward-backwards calculations for hidden Markov models [26 Krogh-94]. Similar calculations are also done using Bayesian inference nets [27 Heckerman-95] [28 Buntine-94] [29 Pearl-88]. Indeed, the Bayesian generalization of the Markov process used by the Tree Model to represent the process of nucleotide duo evolution may be interpreted as a form of Bayesian inference net. In this interpretation, the Bayesian net is given a structure parallel to that of the phylogenetic tree with hidden internal nodes representing the internal nodes of the phylogenetic tree. A detailed derivation and discussion of these calculations is given in [30 Gulko-95]. Given a current estimate of the parameters ρ and φ, we then estimate the frequency with which each type of mutation occurs over the evolutionary reconstruction for each $d \in D_{train}$. The mutation probability matrix ρ is then reestimated by normalizing the mutation frequencies[**], thus providing a new estimate for ρ. This process is iterated until no significant changes in the parameters are observed [31 Thorne-91]. The process may be started with any reasonable initial guess for the parameter values.

The parameters of the model $Model_{nopair}$ were obtained in a manner similar to $Model_{pair}$. The only difference was in the selection of *D*. For $Model_{nopair}$, *D* is a set of column duos, selected at random, from a set of multiple alignment columns which are believed not to contribute to the RNA secondary structure. Once the parameters for both of these models are obtained, they were tested on independent test column duos (D_{test}), not used in the training set. Results of these tests are described further below.

Calculating Likelihood's Using Dynamic Programming

As described above, each *Model* consists of three parts, a classical phylogenetic tree *T*, a mutation probability matrix ρ and an a-priori nucleotide distribution φ. For a given multiple alignment duo *d*, each leaf of the tree corresponds to a particular nucleotide duo determined by that leaf's organism's contribution to the column duo (*Figure 1*). The nucleic makeup of the leaf nodes serves to define the anchor step for a recursive calculation of a nucleotide duo probability distribution for each internal node. The inductive step of the recursion also requires the conditional probability distribution P(*child_node*=*m* | *parent_node*=*l*), where *l* and *m* are nucleotide duos[††]. This conditional probability distribution ($\rho_{l\ m}$) can be interpreted biologically as a mutation rate between nucleotide duo *l* and nucleotide duo *m* over the time span of one phylogenetic tree branch. To make this induction computationally feasible, we additionally assume the standard Markov independence property between a parent node and its immediate descendants, namely,

$$\mathrm{P}(child\_1{=}m \mid parent{=}l \wedge child\_2{=}n\,) \;=\; \mathrm{P}(child\_1{=}m \mid parent\_node{=}l\,).$$

This assumption allows us to define a recurrence relation over the nodes of the binary phylogenetic tree as,

$$\mathrm{P}\big(d\,(parent) | parent = l\big) = ^{\ddagger\ddagger}$$

[**] Since our sample size was fairly large, we did not need to use Bayesian methods in the reestimation of these parameters, as in [26 Krough-94].

[††] Only the 16 nucleotide duos AA, AC, AG, AU, CA, CC, CG, CU, GA, GC, GG, GU, UA, UC, UG, UU are considered valid. Other symbols including ambiguous nucleotides and delete states are not.

[‡‡] In the statement P(*d*(*Node*)|*Node*=*l*), *d*(*Node*) refers not to all of *d*, rather *d*(*Node*) refers to that section of *d* which is descended from *Node* in the phylogenetic tree. As the phylogenetic tree

$$\left[\sum_m \left[\mathrm{P}(d(child\_1)|child\_1 = m)\cdot \mathrm{P}\left(child\_1 = m| parent = l\right)\right]\right]\cdot$$
$$\left[\sum_n \left[\mathrm{P}(d(child\_2)|child\_2 = n)\cdot \mathrm{P}\left(child\_2 = n| parent = l\right)\right]\right] =$$
$$\left[\sum_m \left[\mathrm{P}(d(child\_1)|child\_1 = m)\cdot \rho_{l\,m}\right]\right]\cdot\left[\sum_n \left[\mathrm{P}(d(child\_2)|child\_2 = n)\cdot \rho_{l\,n}\right]\right].$$

The probability P(d(*parent*) | *parent* = l) is analogous to the Inside probability distribution in Lari & Young [25 Lari-90], except here it is applied to a tree shaped Markov Model rather than a Stochastic Context Free Grammar. For a given nucleotide column duo d, we may use this formula to calculate by recursing the calculation from the leaf nodes to the root node at which point we may calculate P(d(*root*)|*Model*) = P(d|*Model*) as,

$$\sum_l \left[\mathrm{P}(d|root = l \wedge Model)\cdot \mathrm{P}\left(root = l| Model\right)\right] =$$

$$\sum_l \left[\mathrm{P}(d|root = l \wedge Model)\cdot \varphi_l\right]$$

Where we have defined our final piece of model φ_l = P(*parent*=l|*Model*). The following example serves to show this process in action.

Paired Model		
P(AU→AU) = .954	P(AU→GC) = .046	P(AU) = .182
P(GC→AU) = .011	P(GC→GC) = .989	P(GC) = .818

Nonpaired (Random) Model		
P(AU→AU) = .975	P(AU→GC) = .025	P(AU) = .361
P(GC→AU) = .027	P(GC→GC) = .973	P(GC) = .639

Table 1: Mutation Model Parameters for Example

These numbers were taken from [30 Brad-95]. The probability of no mutation occurring was maintained and the residual probability assigned to a mutation to the complimentary nucleotide duo. For the *a priori* state distribution, the relative proportions of AU and GC in D_{train} were maintained and scaled up to total 100%.

---

is a binary tree, d(*parent*) = d(*child_1*) ∪ d(*child_2*). It also follows from this definition that d = d(*Root*).

$$\mathrm{P}(d|\mathrm{Model}) = \sum_{l=0,1} \mathrm{P}(d(A_0)|A_0 = l \wedge \mathrm{Model}) \cdot \mathrm{P}(A_0 = l|\mathrm{Model})$$

$$(.0408 \cdot .182) + (.0098 \cdot .818) = .021$$

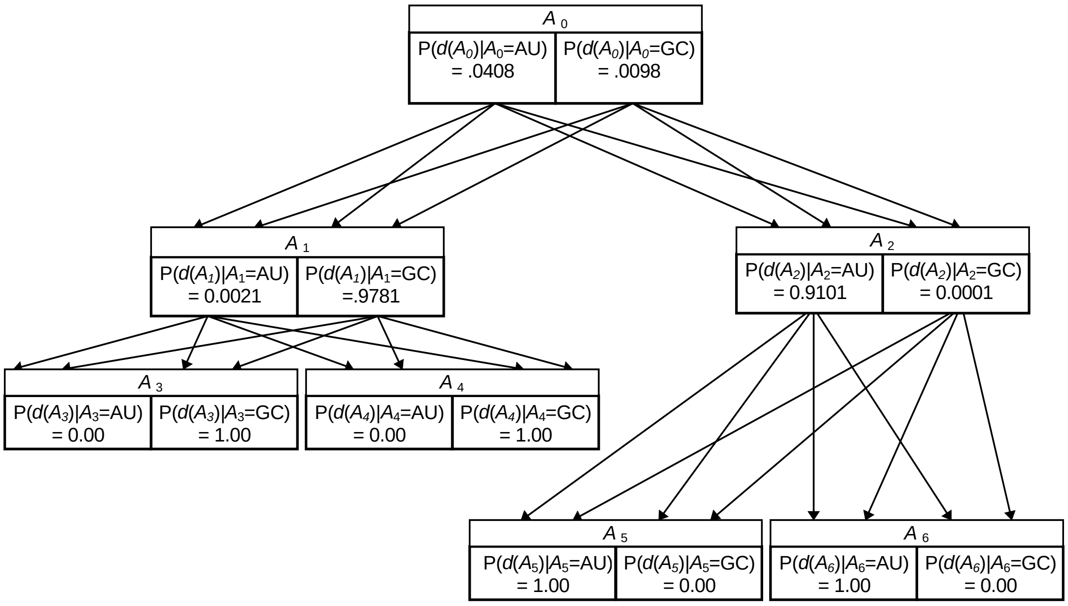


Figure 2: Calculation Tree for Example (Case 1, $Model_{Pair}$)

This tree shows the calculation process used to compute the posterior data probability P(d|$Model_{pair}$) for the column duo d described in *Figure 1*. The leaf nodes are initialized from the known nucleotide duo values from d. Other probabilities are derived from descendants according to the inference equation developed above.

	P(d|Model) for Model Type			
Data d	$\mathrm{Tree}_{\mathrm{Pair}}$	$\mathrm{Tree}_{\mathrm{Rand}}$	$\mathrm{Freq}_{\mathrm{Pair}}$	$\mathrm{Freq}_{\mathrm{Rand}}$
Case 1	.02101	.02364	.05321	.02216
Case 2	.00073	.00066	.05321	.02216

	NNLL(P(d|Model)) for Model Type, (bits/base)			
Data d	$\mathrm{Tree}_{\mathrm{Pair}}$	$\mathrm{Tree}_{\mathrm{Rand}}$	$\mathrm{Freq}_{\mathrm{Pair}}$	$\mathrm{Freq}_{\mathrm{Rand}}$
Case 1	0.697	0.675	0.529	0.687
Case 2	1.302	1.321	0.529	0.687

Table 2: Example Likelihood Result Summary

The NNLL is a Normalized Negative Log Likelihood. Thus NNLL(P) is $\log_2(P)/(2Z)$, where Z is the number of valid nucleotide duos in the column duo. In the present example, $Z = 4$.

Experimental Results

The discriminator described above is tested on data obtained from Ribosomal Data Project [18 RDP-93]. This data contained a multiple alignment [32 RDP-93] of 16S RNA from 1375 organisms, along with an associated phylogenetic tree [33 RDP-93] [34 Olsen-94]. In addition, 472 column duos of known secondary structure were obtained [23 Macke-93] as well as 3500 column duos selected randomly from those known not to be paired. These column duos were then filtered so that at least 75% of the nucleotide duos in each column duo were valid nucleotides duos[††]. This left 317 paired column duos and 695 remaining nonpaired column duos. Each of these data sets was divided into training and validation subsets according to a 4 fold cross validation scheme. The *Model*s were trained and P(*d*|*Model*) was computed for each validation duo under each model. As P(*d*|*Model*) can be on the order of 10^{-1000}, probabilities were converted to a Normalized Negative Log Likelihood (NNLL) form where NNLL(P(*d*|*Model*)) = $-\log_2$(P(*d*|*Model*)) / (2Z), and Z is the number of valid nucleotide duos in column duo (*d*). This method of representing probabilities also has the information theoretic interpretation of bits of information per valid nucleotide. This may be convenient for comparison with mutual information based secondary structure detectors.

NNLL Values				
NNLL	Training Data		Validation Data	
(bits/base)	NoPair Data	Pair Data	NoPair Data	Pair Data
Model(NoPair)	0.313	0.365	0.316	0.364
Model(Pair)	0.495	0.260	0.496	0.283

Table 3: Validation Set NNLL Value Summary

The following tables describe classification accuracy, the first represents the accuracy of a simple comparison classifier. For this classifier, if P(*d*|$Model_{Pair}$) > P(*d*|$Model_{NoPair}$) then multiple alignment column duo *d* is classified as paired secondary structure (*Table 4*). This accuracy is also reflected in *Figure 3*.

Linear Discriminator - Classification Accuracy				
	Training Data		Validation Data	
Predicted	NoPair	Pair	NoPair	Pair
NoPair	7443	153	2462	134
Pair	897	3651	318	1134
Accuracy	91.35%		88.83%	

Table 4: Base Pairing Discrimination Accuracy for Linear Classifier

However, as there is a strong non-linearity of data points near the origin, a non-linear (neural network) classifier was also constructed. As this classifier was trained solely on the *Model* training data, its validation results are reasonable representations of an optimal classifier for the resultant probabilities, yielding a classification accuracy of approximately 91% (*Table 5*).

Neural Net Discriminator - Classification Accuracy				
	Training Data		Validation Data	
Predicted	NoPair	Pair	NoPair	Pair
NoPair	7970	404	2626	224
Pair	370	3400	154	1044
Accuracy	93.63%		90.66%	

Table 5: Base Pairing Discrimination Accuracy for Nonlinear Classifier

The chart in *Figure 3* contains the validation data results in NNLL format. Each *d* is represented by one point on the chart with $P(d|Model_{Nopair})$ along the X axis and $P(d|Model_{Pair})$ along the Y axis. As the data separation between paired and nonpaired data seems to increase with increasing $P(d|Model_{Nopair})$, an accuracy line is provided to help quantify the change. Mutation is a relatively rare event, thus column duos with little change throughout their evolutionary history are given relatively high probability. These duos are difficult to classify because there is no simple way to distinguish a highly conserved paired column from two independently conserved columns. The resolution of this problem is a primary source of ongoing research.

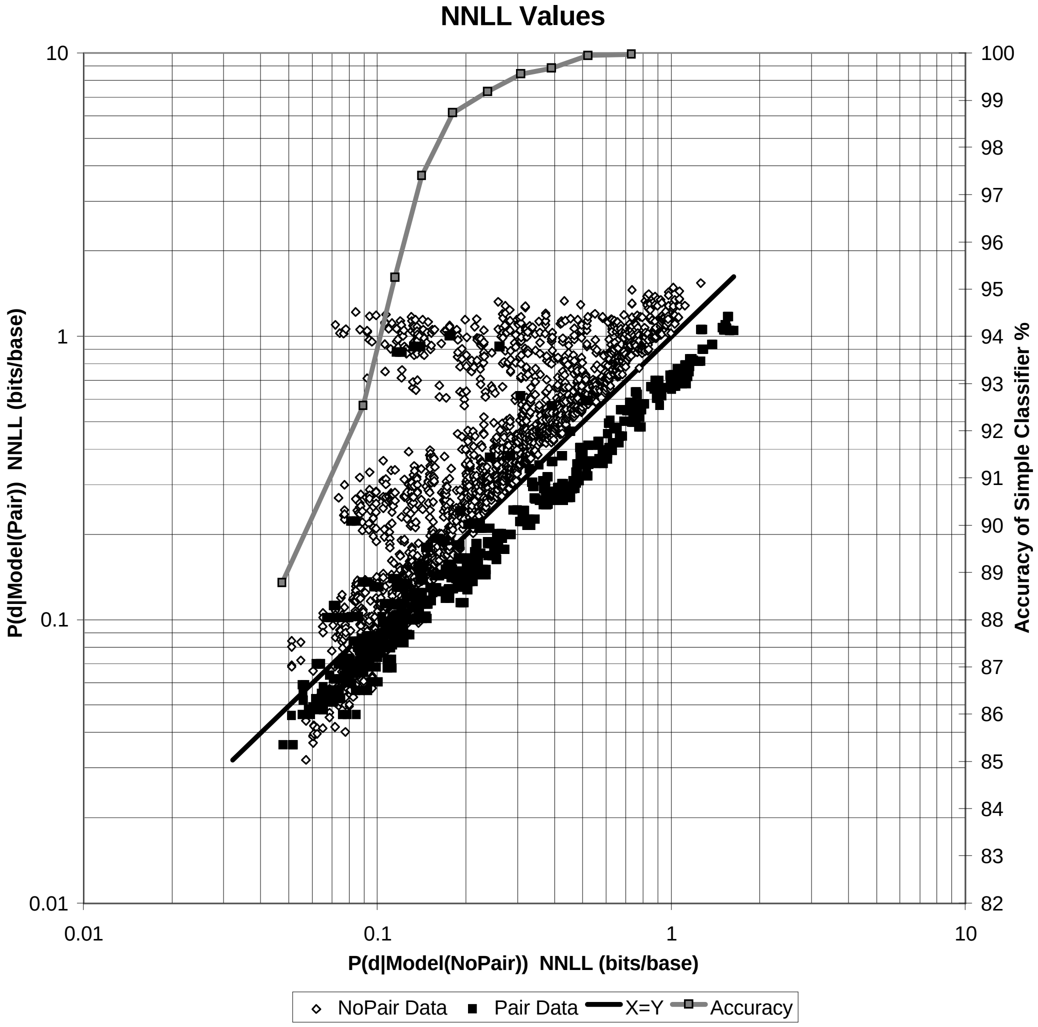


Figure 3: NNLL Values for Validation Data

Each data point above represents a single column duo. Data points noted as Pair Data are drawn from known secondary structure. Data points noted as NoPair Data are drawn from columns which are known to not be paired. For each data point, the X axis value is the likelihood of that column duo, according to $Model_{Pair}$, while the point's Y axis value is its likelihood according to $Model_{NoPair}$. The X=Y line represents the separating boundary for the simple classifier. Points below this line have a higher likelihood of being generated by $Model_{Pair}$ while points above the line have a higher probability of being generated by $Model_{NoPair}$. This discriminator is extremely effective for data which has relatively low likelihood's, but begins to suffer from nonlinearities near the origin. By compensating for the nonlinearities near the origin, the Neural-Network based classifier achieved superior performance. As mutation is a relatively rare occurrence, data with few mutations generally have higher likelihood's. These column duos are more conserved through the evolutionary process. It is these data which are most difficult to classify as it is very difficult to distinguish a highly conserved paired column duo from two highly conserved nonpaired columns. To highlight this phenomena, the Accuracy line displays the increasing resolving power of the simple classifier, as more and more of the conserved column duos are excluded. Each point on the accuracy line represents the cumulative accuracy of the simple classifier for all data to the right of that point. For example, at an X axis value of .2 bits, the simple classifier attains a discrimination accuracy of approximately 98.5% over all data points with NNLL(P($d|Model_{NoPair}$)) > .2 bits. For ease in determining exactly how much data has been excluded at each point on the Accuracy line, a hollow rectangle is placed on the accuracy line for each 10% of the total data points excluded. For example, at X = .2 bits the simple classifier's accuracy is approximately 98.5%, with approximately 40% of the most conserved data excluded.

Comparison of Methods

Despite the relatively high accuracy of the Tree Model in separating the Paired column duos from Nopaired column duos in our sample, two questions remain unaddressed. The first is how does the Tree Model compare with other readily available models on the same population. The second is how does this performance on the sample generalize to the population of all column duos in a multiple alignment?

To answer the first question, we developed a relatively simple mutual information model (*MI*) to test for statistical dependence in the nucleotide duo distribution of the two columns. For each column duo MI calculates a normalized negative log likelihood for the duo under two differing assumptions. The first assumption is that the column duos have a dependent nucleotide distribution. Thus a joint 16 element (4x4) nucleotide duo distribution (φ) is calculated directly from the nucleotide duos found in that column duo. This distribution is then used to calculate the NNLL as:

$$-\sum_{l} \varphi_l \log_2(\varphi_l) / 2$$

This corresponds to P($d|Model_{Pair}$). Under the second assumption we calculate φ as the independent product of each column's individual nucleotide distribution. The NNLL for *d* is then calculated as before using the new φ. This NNLL corresponds to P($d|Model_{Nopair}$). Apart from greater computational efficiency, this method has two advantages over the Tree Model. First, MI takes into account all forms of dependency in nucleotide duo distributions (i.e. GG endcaps), and is not limited to detecting those forms of dependency found in RNA secondary structure. Second, MI uses only information from one column duo at a time, while IOM averages mutation rates over all column duos. It has been shown that multiple alignment column duos evidence differing mutation rates based on their location in an RNA molecule [35 Van de Peer-93] [36 Manske-87]. The inability of IOM to conform to this variance may result in elevated NLL values, and lowered detection sensitivity for the IOM model. As MI calculates statistics separately for each column duo, it can conform to differences between duos.

To characterize the second issue, we note that the test sample of 317 Paired column duos and 695 Nopair column duos does not reflect the general problem of searching for paired column duos in a multiple alignment. In the 16S alignment studied, there were 2688 columns and 472 known paired duos. In a completely general search, we would be looking not for 317 pairs elements from a set of 1,012 (317+695), rather, we would be looking for 472 pairs in a much larger set of 7,222,656 ($2{,}688 \times 2{,}687$) column duos. The scope of our domain is limited somewhat by our data filtering requirement of 75% valid nucleotide duos per column duo, to a search for 317 pairs in approximately 1,400,000 column duos. We also have to contend with an asymmetric utility function, namely, that accurately identifying a few column pairs with high certainty is much more valuable than a marginally higher overall classification accuracy. While we might obtain 99.98% accuracy by merely identifying every column duo as not-paired, such accuracy is of no practical value. To contend with this issue, we use Bayes' Rule to generate posterior model probabilities ($P(Model|d)$) from the likelihood's generated by the Tree Model ($P(d|Model)$) and the prior probabilities $P(Model)$ generated by our overall column duo distribution:

$P(Model_{Pair}) = {}^{317}/_{(1{,}400{,}000+317)}$	≈ 0.000226	From 16S Mult. Align.
$P(Model_{Nopair}) = 1 - P(Model_{Pair})$	≈ 0.999774	Definition.
$P(Model|d) = P(d|Model) \cdot P(Model) / P(d)$		Bayes' Rule.
$P(Model_{Nopair}|d) + P(Model_{Pair}|d) = 1$		Definition.

$$\begin{aligned}P(Model_{Pair}|d) &= P(Model_{Pair}|d)/(P(Model_{Pair}|d)+P(Model_{Nopair}|d)) \\ &= P(d|Model_{Pair})\cdot P(Model_{Pair}) / \\ &\quad (P(d|Model_{Pair})\cdot P(Model_{Pair})+P(d|Model_{Nopair})\cdot P(Model_{Nopair})).\end{aligned}$$

To maximize the expected number of correct classifications, a Bayes Optimal classifier would classify $d \in Model_{Pair}$ iff $P(Model_{Pair}|d) > 50\%$. However, to account for the high cost of false positives in column pair identification, we arbitrarily raise the 50% threshold and observe variations in the percentage of all paired column duos which are correctly classified as the number of incorrectly classified Nopair duos drops. The following chart (*Figure 4*) displays this result.

Here we also display preliminary results for a new model which we will call IOM-2. This model is currently under development by the authors, in conjunction with Gary Stormo, Alan Lapedes and Chip Lawrence. This model begins with a trained IOM model and performs additional Expectation Maximization training of ρ on each column duo, using the aggregate ρ as a prior. This model allows for variations in mutation rates between column duos, while utilizing aggregate statistics over all column duos as a Bayesian prior to limit overfitting.

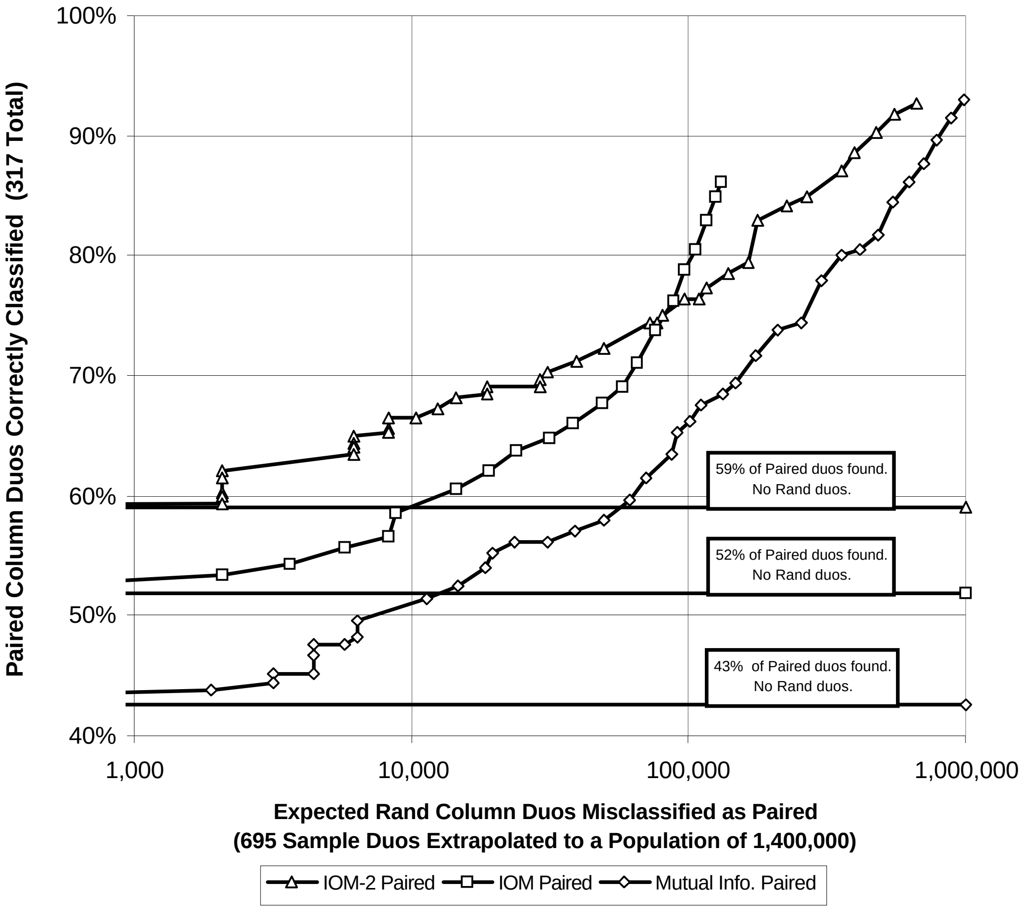


Figure 4: Accuracy of Column Pair Detection Using Posterior Probability

Due to time constraints, our test sample of 695 column duos was randomly selected from the population of 1.4 Million possible non-paired column duos. In the above chart, the X-axis values are scaled by a factor of approximately 2000 to extrapolated algorithmic performance on the entire population of 1.4 Million. All three discrimination methods show a decrease in the number of correctly identified column pairs, as the posterior probability required for classification increases. However, the number of random column duos misclassified as paired drops far more dramatically. Clearly the IOM-2 and IOM methods are superior to the pure mutual information method over a broad range of posterior probability classification thresholds. Explicit classification thresholds are not provided in this chart, though the upper right point of each line represents a 50% probability threshold. Horizontal data lines demark the asymptotic detection percentage of column pairs when non-paired misclassification rates go to 0.

Clearly, the IOM model outperforms MI over a broad range of classification accuracies with the more adaptable IOM-2 model showing even greater selection capability. As the number of random duos misclassified as paired drops asymptotically to 0, the percentage of correctly identified paired duos goes to 59%, 52% and 43% respectively for the IOM-2, IOM and MI models. As each Nopair column duo in our sample represents hundreds of column duos in the population, one might argue that an important subset of 'hard to classify' column duos might be missed. Thus, these precise classification accuracies may be open to argument. Nonetheless, the IOM-2 and IOM models are shown to consistently surpass the Mutual Information model over the sample data and are thus likely to be preferable in practical applications.

Acknowledgments

The authors would like to thank Gary Stormo, Alan Lapedes and Chip Lawrence for several helpful discussions, particularly regarding the IOM-2 Model. We would also like to thank Rodrigo Garces for performing some crucial preparation of the multiple alignment and phylogenetic tree data which was used throughout this work.

References

1 Ann B. Jacobson and Michael Zuker. Structural Analysis by Energy Dot Plot of a Large mRNA. *Journal of Molecular Biology*, 1993, 233:261-269.

2 Kyungsook Han and Hong-Jin Kim. Prediction of common folding structure of homologous RNAs. *Nucleic Acids Research*, 1993, 21(5):1251-1257.

3 Michael Zuker and David Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 1984, 46:591-621.

4 I. Tinoco Jr., O. C. Uhlenbeck and M. D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 1971, 230:363-367.

5 Kyungsook Han and Hong-Jin Kim. Prediction of common folding structure of homologous RNAs. *Nucleic Acids Research*, 1993, 21(5):1251-1257.

6 S. Y. Le and Michael Zuker. Predicting common foldings of homologous RNAs. *Journal of Bimolecular Structure and Dynamics*, 1991, 8:1027-1044.

7 B. D. James, G. J. Olsen and N. R. Pace. Phylogenetic comparative analysis of RNA secondary structure. *Methods in Enzymology*, 1989, 180:227-239.

8 C. Woese, R. Gutell, R. Gupta and H. Noller. Detailed analysis of the higher-order structure of 16S-like ribosomal ribonucleic acids. *Microbiology Reviews*, 1983, 47(4):621-669.

9 R. Cary and G. Stormo, Graph-theoretic approach to RNA modeling using comparative data. *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 1995. Pages 75-80.

10 T. Klinger and D. Brutlag. Detection of correlations in tRNA sequences with structural implications. *First International Conference on Intelligent Systems for Molecular Biology*, 1993, Menlo Park: AAAI Press. L. Hunter, D. Searls and J. Shavlik, editors.

11 R. R. Gutell, A. Power, G. Z. Hertz, E. J. Putz and G. D. Stormo. Identifying constraints on the higher order structure of RNA: continued development and application of comparative sequence analysis methods. *Nucleic Acids Research*, 1992, 20(21):5785-5795.

12 M. S. Waterman. Consensus methods for folding single-stranded nucleic acids. *Mathematical methods for DNA Sequences*, Chapter 8, 1989, CRC Press.

13 F. Lefebvre. An optimized parsing algorithm well-suited to RNA folding. *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 1995, Pages 222-230.

14 S. R. Eddy and R. Durban. RNA sequence analysis using covariance models. *Nucleic Acids Research*, 1994 June 11, 22(11):2079-88.

15 Y. Sakakibara, M. Brown, R. Hughey, I. S. Mian, K. Sjolander, R. Underwood and D. Haussler. Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Research*, 1994 November 25, 22(23):5112-20.

16 Leslie Grate, Mark Herbster, Richard Hughey, David Haussler, I. Saira Mian and Harry Noller. RNA modeling using Gibbs sampling and stochastic context free grammars. *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 1994.

17 Leslie Grate. Automatic RNA secondary structure determination with stochastic context free grammars. *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, 1995. Pages 136-144.

18 Ribosomal Data Project, University of Illinois in Urbana-Champaign. Revision 3.0 of the database. Retrieved from rdp.life.uiuc.edu in /pub/RDP. This source also requests citation of Niels Larsen, Gary J. Olsen, Bonnie L. Maidak, Michael J. McCaughey, Ross Overbeek, Thomas J. Macke, Terry L. Marsh and Carl R. Woese, “The Ribosomal Database Project”, *Nucleic Acids Research*,, 1993, Volume 21 Supplement, pp. 3021-3023.

19 Michael Zuker, Hohn A. Jaeger and Douglas H. Turner. A comparison of optimal and suboptimal RNA secondary structures predicted by free energy minimization with structures determined by phylogenetic comparison. *Nucleic Acids Research*, 1991, 19(10):2707-2714.

20 Jan Pieter Abrahams, Mirjam van der Berg, Eke van Batenburg and Cornelis Pleij. Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. *Nucleic Acids Research*, 1990, 18(10):3035-3044.

21 J. A. Jaeger, D. H. Turner and M. Zuker. Predicting optimal and suboptimal secondary structure for RNA. *Methods in Enzymology*, 1990, 183:281-306.

22 Joseph Felsenstein. A likelihood approach to character weighting and what it tells us about parsimony and compatibility. *Biological Journal of the Linnean Society*, 1981, 16:183-186.

23 T. J. Macke. AE2. Ribosomal Data Project, University of Illinois in Urbana-Champaign. 15, February 1993. The list of base paired column duos was drawn from a data library accompanying the AE2 sequence editor. This data is available through anonymous ftp from rdp.life.uiuc.edu in /pub/rdp/programs/Editor_AE2/ae2.tar.Z. The particular data file used is

found at ae2/lib/paircon.16 in the archive file. The author of this program, T. J. Macke, is reachable at macke@scripps.edu.

24 A. P. Dempster, N. M. Laird and D. B. Rubin. Maximum Likelihood from Incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 1977, 39(1):1-38.

25 K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 1990, 4:35-56.

26 A. Krough, M. Brown, I. S. Mian, K. Sjolander and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 1994, 235:1501-1531.

27 David Heckerman. A tutorial on learning in Bayesian Networks. *Microsoft Corporation, MSR-TR-95-06*. 1995.

28 W. L. Buntine. Operations for Learning with Graphical Models. *Journal of Artificial Intelligence Research*, 1994, Volume 2, Pages 159-225.

29 J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, 1988.

30 Brad Gulko. *Using Phylogenetic Markov Trees to Detect Conserved Structure in RNA Multiple Alignments*, Master's Thesis for the University of California at Santa Cruz board of Computer Engineering, March-1995. Also available by anonymous FTP from ftp.LeptonCorp.com in /pub/compbio/thesis/ths-ps30.ps.gz. .

31 Jeffery L. Thorne, Hirohisna Kishino and Joseph Felsenstein. An Evolutionary Model for Maximum Likelihood Alignment of DNA Sequences. *Journal of Molecular Evolution*, 1991, 33:114-124.

32 Multiple Alignment data is drawn from [18] /rdp/SSU_rRNA/SSU_Prok.gb .

33 Phylogenetic Tree data is drawn from [18] /rdp/SSU_rRNA/tree/SSU_Prok.newick .

34 Gary J. Olsen, Hideo Matsuda, Ray Hagstrom and Ross Overbeek. fastDNAml: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *Computer Applications in the Biosciences*, 1994, 10(1):41-48.

35 Yves Van de Peer, Jean-Marc Neefs, Peter De Rijk and Rupert De Wachter. Reconstructing Evolution from Eukaryotic Small-Ribosomal-Subunit RNA Sequences: Calibration of the Molecular Clock. *Journal of Molecular Evolution*, 1993, 37:221-232.

36 C. L. Manske and D. J. Chapman. Nonuniformity Of Nucleotide Substitution Rates in Molecular Evolution: Computer Simulation and Analysis of 5S Ribosomal RNA Sequences. *Journal of Molecular Evolution*, 1987, 26:226-551.