Towards Modelling Evolution = Mutation modulo Selection in Sequence Alignment.

L. Allison,
Technical Report 95/225,
Department of Computer Science,
Monash University,
Australia 3168.

email: lloyd@cs.monash.edu.au
www: http://www.cs.monash.edu.au/~lloyd/

June 1995

Abstract.

Biological evolution takes place through the mutation of DNA modulated by the pressure of selection which determines which changes are viable. A proposal for modelling both of these factors in the multiple-alignment problem is presented. It is argued that tree-based alignment methods primarily model mutation events, and that most non-tree methods (e.g. profiles) model selection pressures on a family of sequences. A tree-based method specifies mutation probabilities and, if used in isolation, the hope must be that biases due to selection are small or average out. A non-tree method can be used to model the family of sequences - which sequences are fit and viable, and which are non-viable, as members of the family. It is suggested that the two approaches can be combined: the latter can be used to modify the mutation probabilities of the former - to model which mutations are viable and are observed.

Keywords and Phrases: evolution, multiple-alignment, pressure of selection, sequence analysis.

1. Introduction.

It is here suggested that the evolutionary process on coding DNA should be modelled by combining a model of mutation and a model of selection. The *model of mutation* is based on an evolutionary-tree where each internal node represents an ancestral sequence, each leaf node represents a descendant sequence, and each edge represents a period of evolution. The "length" of an edge is defined by the mutation probabilities. An edge applies to one ancestor-descendant pair and its effect is uniform across positions within the two sequences. The *model of selection* is based on a model of the family of proteins that the DNA sequences code for. It takes the form of a profile of the "typical" member of the family, although a hidden Markov model or another kind of model could be used in principle. The profile applies uniformly to all sequences in the family but can have different effects on each position in the sequences. A DNA sequence is as fit as the protein that it codes for. If a certain amino acid in a certain position is lethal then the corresponding DNA sequences are never observed. The two components of the evolutionary model are conventional and have been used elsewhere; their combination seems to be novel.

Mutation models underlie many multiple-alignment methods, particularly those for DNA. Felsenstein (1981) introduced a rates-of-mutation model for recovering an evolutionary-tree from a given multiple-alignment. Bishop et al (1986), Thorne et al (1991) and Allison et al (1990, 1992) applied mutation models to aligning two sequences. Allison and Wallace (1994) used them in Gibbs sampling and simulated annealing of multiple-alignments for a given evolutionary-tree. These methods ignore the pressure of selection (although Thorne et al (1992) modelled variable sites and conserved sites), indeed selection is something of a problem for such methods as it may bias the rates and kinds of mutation that are observed. Pseudo-genes and other sections of so-called junk DNA are often used so as to avoid these difficulties.

Many alignment methods, particularly those popular for proteins, do not use explicit tree-based mutation models at all. Such methods are often based around some kind of model of the "typical" member of a protein family. Such a model might take the form of a profile (Gribskov et al 1991), a block with or without gaps (Henikoff and Henikoff 1991), or a hidden Markov model (Haussler et al (1993)). The instantiated model represents what it is to be a member of the family. A given sequence complies with the model to a greater or lesser extent. The model is often used to screen a data-base of sequences to find new family members. Such models largely ignore the evolutionary history of the sequences, indeed it can be something of a problem for such methods if the object is to understand the function of the sequences rather than the accidents of evolutionary fate. A given alignment may of course reveal conserved regions and hence give clues to function, but the presence of sequences related more and less closely in the data can be a problem to algorithms that do not model this explicitly. Here we argue that non-tree-based models mainly capture the effects of the pressure of selection on the sequences.

Two components, a mutation model and a selection model, should be combined in a true evolutionary model. In order for B to be a descendant not only must A mutate into B but B and the intermediate sequences must be "fit". Therefore the proposed evolutionary model uses the mutation model's probabilities conditional on the selection model. There may be highly probable mutations of A into B', but if B' is unfit it is unlikely to evolve and be observed as a descendant.

The same philosophy could be applied to tRNA which is also subject to a pressure of selection.

2. Mutation Models.

A mutation model over several sequences specifies an evolutionary-tree. The given sequences are the leaves of the tree. The hypothetical ancestral sequences are the internal nodes of the tree. Binary trees are most often used, but n-ary trees can be used. The trees can be rooted or unrooted; unrooted trees are used here, but this is partly a matter of taste. The longer an edge of the tree, the more time it represents, and the higher the probabilities of mutations on that edge.

Mutations are assumed to be rare events. Given the four sequences S1=S2=ACT and S3=S4=AGT (figure 1) and assuming that all edges in a tree are of equal length, the tree ((S1 S2) (S3 S4)) is more likely

```
((S1 S2) (s3 S4))                   ((S1 S3) (S2 S4))

                  S1 = ACT                              S1 = ACT
  ACT                                ACT or AGT
                  S2 = ACT                              S3 = AGT

                  S3 = AGT                              S2 = ACT
        AGT                                   ACT or AGT
                  S4 = AGT                              S4 = AGT

  >=1 mutation                          >=2 mutations

(edges not to scale)
```

Figure 1: Two Evolutionary Trees over S1-S4.

than ((S1 S3) (S2 S4)) because the former requires one mutation and the latter requires two mutations.

The model of character substitution, insertion and deletion can be simple or complex and is not a focus of this paper (it is a parameter of the method being described). In the simplest models, all substitutions occur at the same rate and their probabilities increase together with time. More complex models allow transitions and transversions different rates, or even allow each kind of substitution its own rate. Inserts and deletes (indels) can be given separate rates or can be dealt with in various ad-hoc ways.

An unrooted binary tree over K leaves has 2K-3 edges and there is at least one parameter for each edge (its "length") which must be inferred from the data. There may be more parameters in the more general models. (There are 2K-2 edges in a rooted tree and if a constant rate of mutation is assumed then there are only K-1 independent lengths.)

A mutation model, as described, treats all positions in the sequences equally. For example, there is no way to specify that position 1 is highly conserved and that changes in position 2 are much more probable. (Thorne et al (1992) did augment a mutation model with a new component which specified conserved and variable regions and which was inferred from the data.) We propose augmenting a mutation model with a selection model of the typical member of the sequence family.

3. Selection Models.

Many workers in proteins are more interested in how a protein works, which parts are conserved, where the active site is and so on, rather than its evolutionary history. This is not to say that the history is of no interest or can throw no light on function, it just may be of less interest to them. A model of the family of sequences or of a typical member of the family is usually built in this case. The model can be implicit - a multiple-alignment for example - or explicit; we consider multiple-alignment based on all-pairs costs and on star costs here.

A star relates every given sequence to a single central sequence, S (e.g. figure 2). It is arguably a simple mutation model if S is taken to be the original ancestral sequence. All edges typically have the same length, although this can be generalized. The aim is to find S that optimizes a cost or score, summed over S being compared to each leaf sequence. A star's strong point is also its weak point - it avoids any question of the correct tree topology. S may approximate the typical family member provided that the true evolutionary-tree does not bias the inference.

```
S1=ACT            S4=AGT
      .          .
        .      .
         S= ACT
         or AGT
        .      .
      .          .
S2=ACT            S3=AGT
```

Figure 2: Star over S1-S4 Requires >= 2 Mutations.

All-pairs costs score a multiple-alignment as the sum over all pairs of given sequences (e.g. figure 3). This avoids the introduction of any unknown (ancestral) sequences, even the single sequence introduced by a star.

A frequency matrix or ‘profile’ (Gribskov et al 1991) is a simple, explicit model often built from a multiple-alignment; an example on DNA is given in figure 4. A profile consists of a probability distribution of letters, possibly including the null or gap character, for each position in the alignment. Each column can be modelled by a multistate distribution (Wallace & Boulton 1968, Boulton & Wallace 1969). A profile can be formed from a multiple-alignment by counting the frequencies of letters in each column; note that a count of zero for a letter should be interpreted as a low but non-zero probability. This assumes that the given sequences are representative of the family. The inferred profile may not accurately reflect the family if, for example, the data contains a large closely related subfamily which can bias the statistics in its direction.

A profile without gaps is often called a block. A model might consist of a number of blocks (separated by gaps).

A hidden Markov model (HMM) is a more complex explicit model. A HMM consists of a number of states connected by transition arcs. As used by Haussler et al (1993) there are delete, insert and match states for each position in the protein. A match state generates a letter, according to a probability

```
S1=ACT-------------S4=AGT
    . .            . .
    .   .        .   .
    .     .    .     .
    .       .        .
    .     .    .     .
    .   .        .   .
    . .            . .
S2=ACT-------------S3=AGT

Sum costs over all pairs = 4
```

Figure 3: All-pairs Costs over S1-S4.

```
sequences S1 = ACT
          S2 = ACT
          S3 = AGT
          S4 = AGT
Profile    A : 400
           C : 020
           G : 020
           T : 004
```

Figure 4: Profile over S1-S4.

distribution. A delete state skips a letter. An insert state inserts a letter in a sequence. The most probable path(s) through the HMM represent typical family members. The architecture of a HMM can be varied to suit an application. A profile can be thought of as a simple HMM with only match states.

We argue that all of the above principally capture information about fitness and the pressure of selection on the family of sequences - what it is to be a successful member of the family. Indeed, once they have been instantiated, such models are sometimes used to search sequence databases to find new candidate members of the family. They can also be used to estimate the fitness or functionality of artificial sequences. Such models are very simple and only capture constraints on individual amino acids at each position in a protein. In particular they ignore pair-wise constraints between amino-acids that become close when the protein folds. Nevertheless such models are widely used and useful. We propose that such a model be used to model explicitly the pressure of selection on given sequences and hence call them selection models.

4. Mutation modulo Selection.

The use of a mutation model M to model the evolutionary history of given coding DNA sequences and the use of a selection model N to model the pressure of selection on them is proposed. For example, a DNA sequence S may code for a protein P. S may mutate into S' which codes for P'. If P' scores highly under N then S' is likely to survive and be observed (be present in the given data) or have descendants which are observed. If P' does not score highly, it is not fit and S' is likely to die out unobserved. (It might be claimed that that this is not the way that selection works, but we counter that it is at least a step in the right direction and that without a biochemical theory of how the proteins work it is a good place to start.) In order for S' to be observed it must first arise by mutation and then it must be fit enough, or lucky enough, to survive.

```
P = .... Q ....
S = ....pqr....

P’= .... X ....
S’= ....pxr....  where x in {ACGT}
```

Figure 5: Single Letter Change.

We use the convention that the codon abc codes for amino acid B or B', def codes for E or E', etc. in what follows. We begin by considering the case of at most one base change in a single codon. For example, the codon pqr which codes for the amino acid Q mutates to pxr which codes for X (figure 5); the change might be in any position in the codon, Q might or might not be the same amino acid as X, and x could equal q. The probability of this mutation is denoted Pm(pqr->pxr in time T) or Pm(pqr->pxr) if the time is assumed from context. The probability that pqr evolves into pxr and that the change is viable and is observed is denoted Pe(pqr->pxr). Ps(X) is the probability of amino acid X being in a typical fit sequence. This is *not* in general the proportion of *given* proteins that contain X; it is the (inferred) probability of X being in an arbitrary fit member of the protein family. Point mutations are assumed to be independent.

```
(i)   Pm(pqr->pxr) = P(p->p).P(q->x).P(r->r)

(ii)  Ps(Q) = P(Q | member of protein family)

                       Pm(pqr->pxr).Ps(X)
(iii) Pe(pqr->pxr) = ---------------------------
                       sum[stu] Pm(pqr->stu).Ps(T)

where the sum is over stu in {yqr,pyr,pqy} for y in {ACGT}
Similarly for pqr->pqx, pqr->xqr etc.
```

Imagine pqr mutating into many descendants, pqa, pbr, etc. in proportion to the mutation probabilities. One descendant is then chosen at random or observed where the chance of a particular descendant being chosen (surviving, being fit) is proportional to the probability of that descendant coding for an amino acid of the family. The fitter mutations have more chance of being observed. For example, if X is lethal then Ps(X)=0 and the mutation is not observed. If all amino acids are equally suitable then the probability of *observing* a mutated codon is proportional to the probability of the mutation occurring. If all P(a->b) including P(a->a) are equal, the probability of a certain amino acid being expressed by the mutated sequence is proportional to the number of codons for that amino acid that can be reached in at most one mutation event.

The effect of equation (iii) is that Pe(c1->c2) is Pm(c1->c2) multiplied by a constant that depends on the codons and c1's position in the coding DNA sequence. The underlying mutation rate is not altered but there can be a bias towards codons for a group of amino acids, or even a single amino acid, and this can change the observed mutation rate in that position. Only the relative values of the Ps() are important and normalizing these to sum to one gives a natural interpretation of protein profiles.

Next consider the case of a double base change in a single codon. This is more complex because the mutation presumably took place in at least two stages (e.g. figure 6) and the intermediate codons might express different amino acids with different fitnesses. The simplest approach is to ignore the intermediate codons and their fitness and consider only the initial and final codons pqr and pxy, say, and their products. A better approach is to consider at least the more probable 'path', ie. either pqr->pxr->pxy or pqr->pqy->pxy. The best approach is to sum the probabilities of both paths:

```
(iv) Pe(pqr->pxr->pxy in time T)
     = INTEGRAL[t=0..T] Pe(pqr->pxr, time t).Pe(pxr->pxy, time T-t)

(v)  Pe(pqr->pxy time T)
     = Pe(pqr->pxr->pxy time T) + Pe(pqr->pqy->pxy time T)

(and similarly for any other double change)
```

There is actually an infinite number of paths by which pqr can evolve into pxy. The same is true of

```
          P = .... Q ....
          S = ....pqr....
                .    .
              .        .
            .            .
P1= .... Q'....     P2= .... X'....
S1= ....pqy....     S2= ....pxr....
            .            .
              .        .
                .    .
          P'= .... X ....
          S'= ....pxy....
```

Figure 6: Double Letter Change.

pqr->pxr, e.g. pqr ->pqy ->pxy -> pxr, and even pqr->pqr, e.g. pqr ->pxr ->pqr. There remains a very large number of acyclic paths if loops, in which a codon is repeated, are ignored. It is infeasible to consider more than the direct paths. Doing so, there must be a small chance of ignoring a good indirect path through fit intermediate codons.

Changes in all three letters of a codon imply at least two intermediate steps (e.g. figure 7). As before, the intermediate steps could be ignored, or (better) the best path considered, or (best) the sum taken over the probabilities of all six direct paths.

```
                    P = .... Q ....
                    S = ....pqr....
                  .          .      .
                .            .        .
              .              .          .
P1= .... Q'....     P2= .... Y'....     P3= .... Q"....
S1= ....xqr....     S2= ....pyr....     S3= ....pqz....
         .  .            .         .            .  .
         .    .     .                .     .       .
         .       .                      .          .
         .    .     .                .     .       .
         .  .            .         .            .  .
P4= .... Y"....     P5= .... Q"'...     P6= .... Y"'...
S4= ....xyr....     S5= ....xqz....     S6= ....pyz....
              .              .          .
                .            .        .
                  .          .      .
                    P'= .... Y ....
                    S'= ....xyz....
```

Figure 7: Triple Letter Change.

For any ancestor codon abc and descendant codon def, the selection model modifies the pure mutation probabilities P(abc->def) to give evolution probabilities Pe(abc->def).

Figure 8 illustrates one effect of combining models of mutation and selection. Given an alignment of four proteins (under either tree 1, or under star or all-pairs costs) that aligned the amino acids AABB in a particular column, this would be (some) evidence that A and B are equally fit and fitter than any other amino acid, in that position. Given the same alignment under tree 2, this would be (some) evidence that B is fitter than A because the presence of two As may only be due to the short lengths of the edges leading into them. Tree 2 is approximated by tree 2b which implies that B is probably the fitter. In other words, if a family of proteins contains a large but *closely related* subfamily, that subfamily can no longer bias the description of the typical (fit) member of the family in its direction because the mutation model will explain the effect away.

Finally, a DNA coding sequence consists of many codons and its protein consists of many amino acids. There are likely to be changes to several codons between a sequence S and a descendant S'. This leads to a combinatorial number of intermediate sequences and possible paths. Following common practice in alignment, sequences S and S' are compared on an element by element, ie. codon by codon, basis. The -log probabilities of codon evolution, Pe(), with mutation modified by selection as above, are used as costs in the alignment process.

```
A                        B                          B                     B
  .                    .                          .                     .
    .                .         A                .                     .
      .            .             .            .                     .
        ..........                 ..........           A..........
      .            .             .            .                     .
    .                .         A                .                     .
  .                    .                          .                     .
A                        B                          B                     B
          tree 1                    tree 2                tree 2b

sequences S1 = ...A...
          S2 = ...A...
          S3 = ...C...
          S4 = ...C...
Profile    A : ...2...
           B : ...2...
           C : ...0... etc.
```

A, B: amino acids at corresponding positions in different proteins.

Figure 8: Three Trees.

5. Estimation.

The previous section assumed that the mutation model and the selection model were known, but in general they must be inferred from the data. The mutation model Pm() and the selection model Ps() give the observed mutation probabilities Pe() per position. Pe() yields the probabilities of various tuples (of codons) in a multiple-alignment following Felsenstein (1981). The -logs of tuple probabilities are used as *costs* of tuples in multiple-alignment. If working from a *given* multiple-alignment, Pm() and Pe() could be optimized to maximize the likelihood (minimize the cost) of that alignment. This could be done for two or more given multiple-alignments to determine which is best.

```
--> Pm()->-|
|          |-> Pe()--> tuple prob's/pos'n--> mult' alignment-->-|
|-> Ps()->-|                                                    |
|                                                               |
|---------------<-----------------------------------------------
```

The problem is harder if no multiple-alignment is given and a good one must be found - Pe() determines the likelihood of an alignment and the alignment determines the parameters of Pm() and Ps(). An expectation-maximization approach (Dempster et al 1977) could be used: "bland" initial models assumed, a multiple-alignment found, the parameters of the models reestimated and the process iterated until convergence. Unfortunately the selection model is dependent on the position or tuple-number within the alignment and the length of the alignment will in general vary slightly from iteration to iteration. Codons in one sequence may move relative to codons in another sequence. It may be unclear which tuple in the alignment a codon from a sequence should belong to and how many tuples there are, in other words how a selection model derived from one alignment can be used on another. Various ad-hoc solutions may be possible. One is to attach the tuple-number (from the previous alignment) to each codon in each sequence. Codons could "vote" on the old tuple-number, ie. the position in the previous Ps() to be used, when they were combined during alignment, conflicts being resolved stochastically or by averaging. Another solution would involve forcing all alignments to have the same length by permitting all-null tuples and providing an "adequate" number of them initially. In any case, a stochastic search (Allison & Wallace 1994) for a multiple-alignment would be necessary to avoid becoming stuck in local optima.

Initial values for the mutation model could be set at any reasonable mutation probabilities or based on favourite alignment "weights" (Allison 1993). There are a number of "bland" initial selection models. One would have each amino acid equally fit in each position. A second would have the fitness of an amino acid proportional to the number of codons for that amino acid. Variations could take codon usage into account.

6. Conclusions.

A mutation model based on an evolutionary-tree can give biased estimates in the presence of a pressure of selection. It has been argued that profiles, blocks, HMMs and the like mainly model selection pressure on a family of proteins. Such a selection model can be used to modify the mutation probabilities of a mutation model in a step towards a true evolutionary model. The method has not been implemented to date.

7. Acknowledgements.

Thanks go to Joe Felsenstein for discussing evolution and variable and conserved sites, to Charles Lawrence for a lively discussion on blocks and selection, and to Jeff Thorne for comments on early sketches of this work. The world wide web page for related work has the URL
http://www.cs.monash.edu.au/~lloyd/tildeStrings/

8. References.

Allison L. Normalization of affine gap costs used in optimal sequence alignment. J. Theor. Biol. **161** 263-269 1993.

Allison L. & C.S.Wallace. The posterior probability distribution of alignments and its application to parameter estimation of evolutionary trees and the optimization of multiple alignments. J. Mol. Evol. **39** 418-430 1994.

Allison L., C.S.Wallace & C.N.Yee. Inductive inference over macro-molecules. AAAI Symposium on Artificial Intelligence and Molecular Biology, Stanford, 5-9 1990.

Allison L., C.S.Wallace & C.N.Yee. Finite state models in the alignment of macro-molecules. J. Mol. Evol. **35** 77-89 1992.

Bishop M.J. & E.A.Thompson. Maximum likelihood alignment of DNA sequences. J. Mol. Biol., **190** 159-165 1986.

Boulton D.M. & C.S.Wallace. The information content of a multistate distribution. J. Theor. Biol. **23** 269-278 1969.

Dempster A.P., N.M.Laird & D.B.Rubin. Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. B **39** 1-38 1977.

Felsenstein J. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. **17** 368-376 1981.

Gribskov M., R. Luthy & D. Eisenberg. Profile analysis. Methods in Enzymology **183** 146-159 1991.

Haussler D., A.Krogh, S.Mian & K.Sjolander. Protein modelling using hidden Markov models: analysis of globins. 26th Hawaii Int. Conf. Sys. Sci. **1** 792-802 1993.

Henikoff S. & J.G.Henikoff. Automated assembly of protein blocks for database searching. Nucl. Acids Res. **19** 6565-6572 1991.

Lawrence C.E., S.F.Altschul, M.S.Boguski, J.S.Liu, A.F.Neuwald & J.C.Wooton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science **262** 208-214 1993.

Thorne J.L., H.Kishino & J.Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. J. Mol. Evol. **33** 114-124 1991.

Thorne J.L., H.Kishino & J.Felsenstein. Inching toward reality: an improved likelihood model of sequence evolution. J. Mol. Evol. **34** 3-16 1992.

Wallace C.S. & D. M. Boulton. An information measure for classification. Computer J. **11**(2) 185-194 1968.