Chapter 4

`postal`: Software for Checking Multiple Alignment Accuracy

4.1 Introduction

The idea of site-to-site reliability indicators for pairwise and multiple alignments is not a new one. Several such indicators have been proposed, ranging from residue conservation at a site, through sliding-window and exclusion [MV96] techniques to the fully probabilistic [Miy94, ZLL97]. There are many potential uses for a good reliability indicator; in addition to providing information that could help interpret alignments, such an indicator could be used to identify regions of a pairwise or multiple alignment that may be poorly aligned, thus providing further assistance to the sequence analyst.

As projects to classify proteins attempt to keep up with the expansion of databases, such automated sanity checks turn from luxuries to necessities. Release 3.1 of the Pfam database contains 1313 multiple alignments, each representing a protein domain [SEB+98]. Inspecting all these alignments for errors by eye is unfeasible and there is a clear need for automation. Recent efforts to establish a probabilistic basis for sequence alignment suggest posterior probabilities as a natural way of estimating alignment reliability [Miy94, ZLL97, DEKM98, Kro94, BH96, HD98]. Motivated by this, new software has been developed to check multiple sequence alignments for suspicious regions using posterior probabilities as alignment accuracy indicators.

In this chapter the mathematics of posterior probability are first reviewed. A new software tool - `postal` - based on the HMMER2.0 distribution [Edd95], that displays site-to-site posterior probabilities for multiple alignments and flags low-scoring regions for special attention, is then presented. The software is evaluated by running it on the October 1998 release of Pfam and assessing the pathology of the candidate misaligned regions that the program picks out. Further potential applications of Bayesian methods in sequence alignment are discussed.

4.1.1 Mathematical overview

In the probabilistic view, the score of an alignment $\mathbf{a}$ between a set of sequences $\{\mathbf{X}\}$ is proportional to the log of $\Pr[\mathbf{a}, \{\mathbf{X}\}]$, the likelihood of that alignment under some model that represents our assumptions about the way sequences evolve. (For example, the model might be that "pairs of related protein sequences have local regions of homology, with randomly scattered indel events and independently distributed patterns of amino acid substitution"; this contains the assumptions of the Smith-Waterman algorithm.) The likelihood $\Pr[C, \{\mathbf{X}\}]$ of a particular alignment segment C (such as, for example, an individual residue pair in a Smith-Waterman alignment) can be found by summing the likelihoods of all alignments that include that segment (i.e. $\Pr[C, \{\mathbf{X}\}] = \sum_{\mathbf{a}:C\in\mathbf{a}} \Pr[\mathbf{a}, \{\mathbf{X}\}]$), and the *posterior probability* $\Pr[C|\{\mathbf{X}\}]$ of the segment C is found by dividing the likelihood of C by the total likelihood of all possible alignments (i.e. $\Pr[C|\{\mathbf{X}\}] = \Pr[C, \{\mathbf{X}\}] / \sum_{\mathbf{a}} \Pr[\mathbf{a}, \{\mathbf{X}\}]$). This quantity $\Pr[C|\{\mathbf{X}\}]$ is the desired reliability indicator for the segment C.

This may be illustrated with a concrete example. Suppose one has a hidden Markov model (HMM) profile of a multiple alignment and a sequence $\mathbf{X}$ that one wants to fit to that profile. Suppose further that one wants to assess the evidence for whether position i of the query sequence is aligned to a particular state j in the profile, representing a site of interest. Begin by laying out the HMM profile and the query sequence on the vertical and horizontal axes (respectively) of a dynamic programming matrix (Figure 4.1). The alignment $(i \diamond j)$ of residue i to site j corresponds to the cell marked C in the matrix. To find the likelihood $\Pr[C, \mathbf{X}]$ that i is aligned to j, one must compute the sum of the likelihoods $\Pr[\mathbf{a}, \mathbf{X}]$ of all alignment paths $\mathbf{a}$ that run from the top left corner of the matrix through cell C and on to the bottom right corner. The algorithm for calculating the sum of alignment likelihoods is very similar in appearance to the Viterbi algorithm for calculating the highest-scoring alignment. Since alignment scores are additive, the top-left and the bottom-right quadrants can be treated as

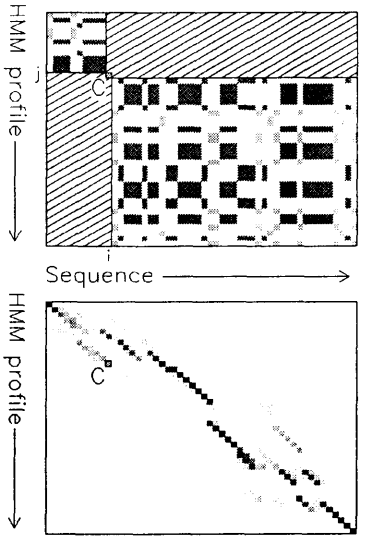

Figure 4.1: The dynamic programming matrix representation of the Forward-Backward algorithm. The Forward likelihood-sum-over-alignments (corresponding to paths spanning the upper left quadrant) is multiplied by the Backward sum (the lower right quadrant) to find the likelihood of all alignment paths passing through the cell C.

independent global alignments and the likelihood sum for each quadrant can be found separately. The likelihoods can then be combined to find $\Pr[C|\mathbf{X}]$. This procedure is known as the Forward-Backward algorithm and is described in more detail in Chapter 2.

For any sufficiently simple (i.e. Markov) model, $\Pr[C|\{\mathbf{X}\}]$ may be calculated using the procedure described above. It has been shown to recover the true probability distribution of C in simulations using a simple Needleman-Wunsch model (see Chapter 2 and [HD98]). From a Bayesian viewpoint, it makes more sense to work with the full posterior distribution $\Pr[C|\{\mathbf{X}\}]$ than to throw out all the cells C that aren't in the optimal alignment; the latter tactic is analogous to making over-precise numerical measurements without taking account of experimental errors.

Ideally, the dynamic programming approach described above would be car-

ried over to simultaneous multiple alignment of many sequences. However, the size of the dynamic programming matrix scales geometrically with the number of sequences being considered and, while various heuristic methods may be employed to home in on the most likely alignment [HBF92, Edd95, NH96, LAB+93] an application of these methods towards estimation of the sums of likelihoods of many alignments remains appealing but untried. Alignment of multiple sequences to ready-made profile HMMs, on the other hand, scales much better computationally and requires no approximations or guesswork (other than during the construction of the HMM itself), which makes finding posterior probabilities for profile-based alignments that much easier. The software presented here uses the profile approach.

4.2 The `postal` software

The `postal` program builds a HMMER profile from a multiple alignment using C functions from the HMMER package [Edd95]. For each sequence in the multiple alignment, it calculates all the posterior probabilities along the alignment path of that sequence to the profile (this alignment is known, since it was used to construct the profile in the first place). The original multiple alignment is output together with single-digit annotation (indicating the first digit of the posterior probability for each site) in the `MUL` format for the `BELVU` program [SD94]. This format can also be read - and attractively displayed using colour - by the `jalview` Java multiple alignment viewer [Cla98].

The `postal` program has a number of options for advanced usage. For example, it can attempt to improve the multiple alignment (see also Section 4.2.3) or it can write the posterior probability tables directly to a file to be read by other programs. An algorithm utilising `postal` probabilities is in development at the time of publication [Gol98].

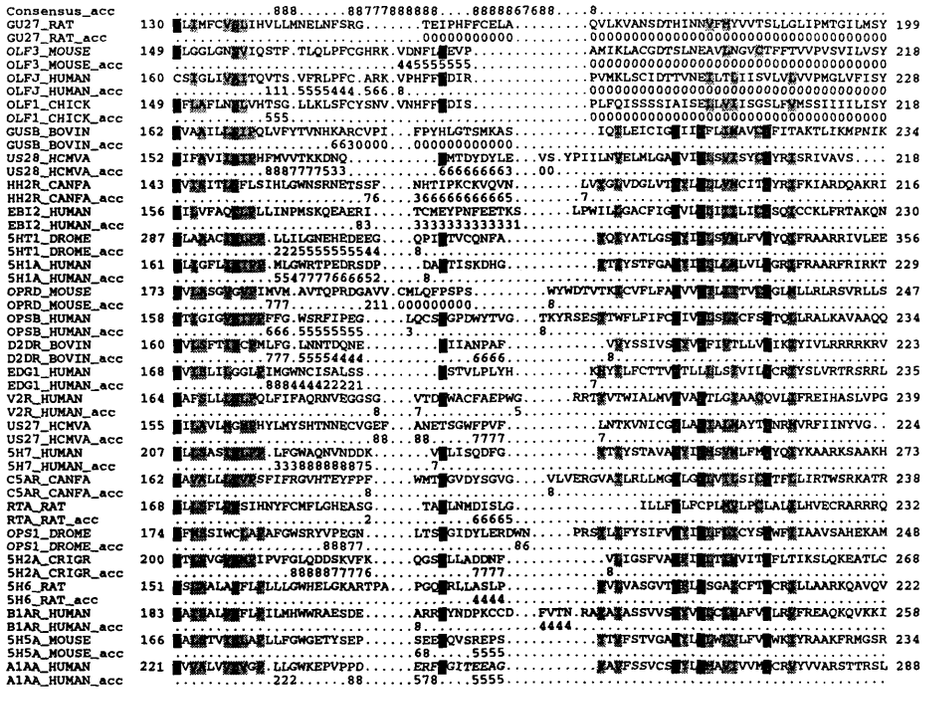

Figure 4.2: Accuracy levels for the 7tm_1 rhodopsin-like domain from Pfam (accession number PF00001). The sequences are sorted with the most suspicious at the top. The block of mammalian olfactory receptors with "0" accuracy on the top right-hand side are misaligned; to see this, note that the rest of the alignment has a column of conserved tyrosine residues aligned to residue 185 of the top sequence, while the misaligned block has a corresponding column of tyrosines at residue 199 of the top sequence. In other parts of the alignment, weak matches near gapped regions are often seen to have low probability.

4.2.1 Usage

Figure 4.2 shows an example of the output of `postal`, plotted by the `Belvu` multiple sequence alignment viewer [SD94]. The displayed alignment is part of the 7tm_1 rhodopsin-like domain from Pfam (accession number PF00001) - the same alignment as in Chapter 1, Figure 1.1. The aligned sequences are sorted with the most suspiciously aligned at the top. Beneath each sequence is an accuracy line, with the digits `0-9` indicating confidence levels for each site (low numbers signify low-accuracy regions, with a '9' indicating predicted perfect accuracy, a '5' indicating ambiguity and a '0' indicating that HMMER would rather put that residue with a different column). The top line is a "consensus accuracy" line obtained by averaging all the accuracy levels in a column, if the column comprises less than $\frac{2}{3}$ gap characters. By default, the (usually prevalent) digit '9' is masked out with dots to make suspicious regions easier to pick out. This alignment contains several suspicious regions, including one section that is clearly misaligned and several others where the column conservation is poor (see figure legend).

4.2.2 A note on interpretation

The posterior probabilities described here denote the confidence of the alignment model in a particular alignment. A low probability indicates ambiguity as to how a particular residue should be aligned. This is due to the absence of a strong signal, maybe because the sequence has little information content in this region, or because there are a lot of gaps nearby or even because the HMM training method is flawed. A high probability means that a sequence is well anchored, though not necessarily prettily aligned. For example, a run of mismatches sandwiched in the middle of an ungapped block will often have a high probability if the flanking sequences match the block consensus (though this may also depend on the gap-insertion policy of the algorithm). To take another example, given a handful of unrelated sequences, it is usually possible to

train a probabilistic model to recognise these sequences well, despite the lack of homology between them. Assuming the sequences have any information content whatsoever, the (trained) model will then assert that the posterior probabilities of the training sequences are close to 1. In other words, probability theory is only as good as the underlying assumptions; the use of posterior probabilities may reveal ambiguities in an alignment, but without making new assumptions one may not be able to detect all the sequences that are badly aligned.

4.2.3 The optimal accuracy algorithm and `postal`

The `postal` program implements the optimal accuracy algorithm described in Chapter 3 for HMM profile alignment. This feature remains experimental and has not been systematically evaluated for HMM profiles.

4.2.4 More complex models

At the core of the `postal` software is the program `hmmbuildpost`, which calculates and prints a table of posterior probabilities for the alignment of a single sequence to a profile HMM. In addition, it finds the Viterbi and optimal-accuracy alignments of the sequence to the HMM (the optimal-accuracy algorithm finds the alignment $\mathbf{a}$ that maximises the expected overlap score $E(|\mathbf{a} \cap \mathbf{a}_{\mathbf{real}}|) = \sum_{C \in \mathbf{a}} \Pr[C|\{\mathbf{X}\}]$; see Chapter 3 or [HD98] for an evaluation of this algorithm). The `hmmbuildpost` program is transparently invoked by `postal` and should rarely need to be used on its own.

The `hmmbuildpost` program has the same output format as the `modelpost` program, a more general tool for working with posterior alignment probabilities and optimal-accuracy alignments that is independent of the HMMER package. Details of this program are available on the `postal` web site.

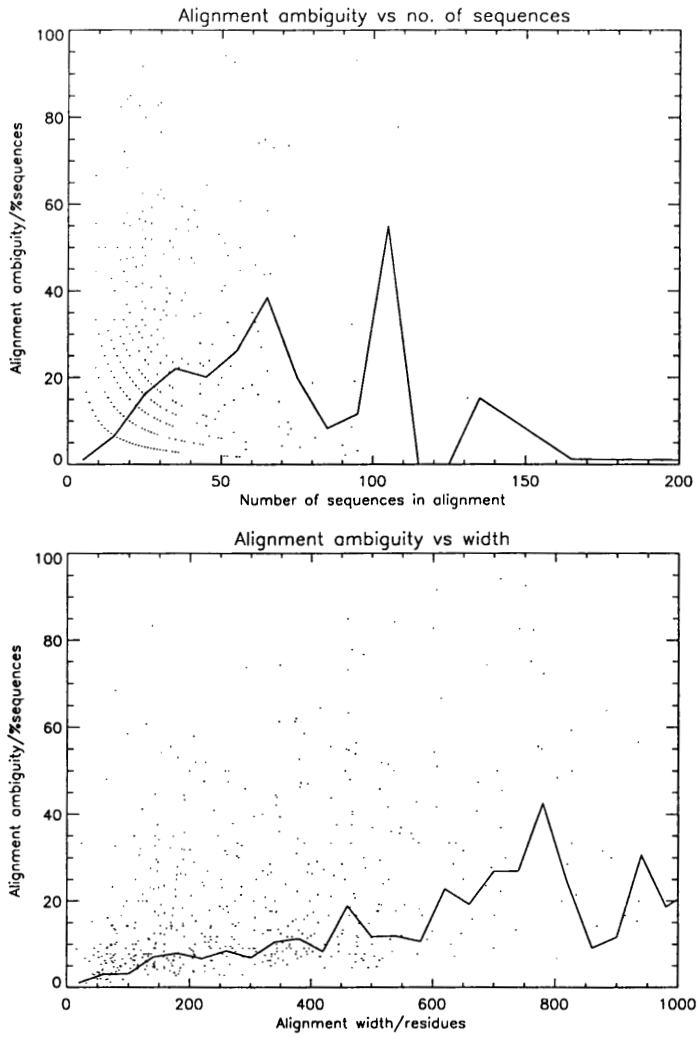

Figure 4.3: The two scatterplots show the proportion of sequences containing ambiguous regions, plotted against the total number of sequences in the alignment (upper plot) and the alignment width (counting both residues and gaps) (lower plot). There is a direct correlation between alignment ambiguity and alignment size. The solid lines represent y-averages for binned x-values.

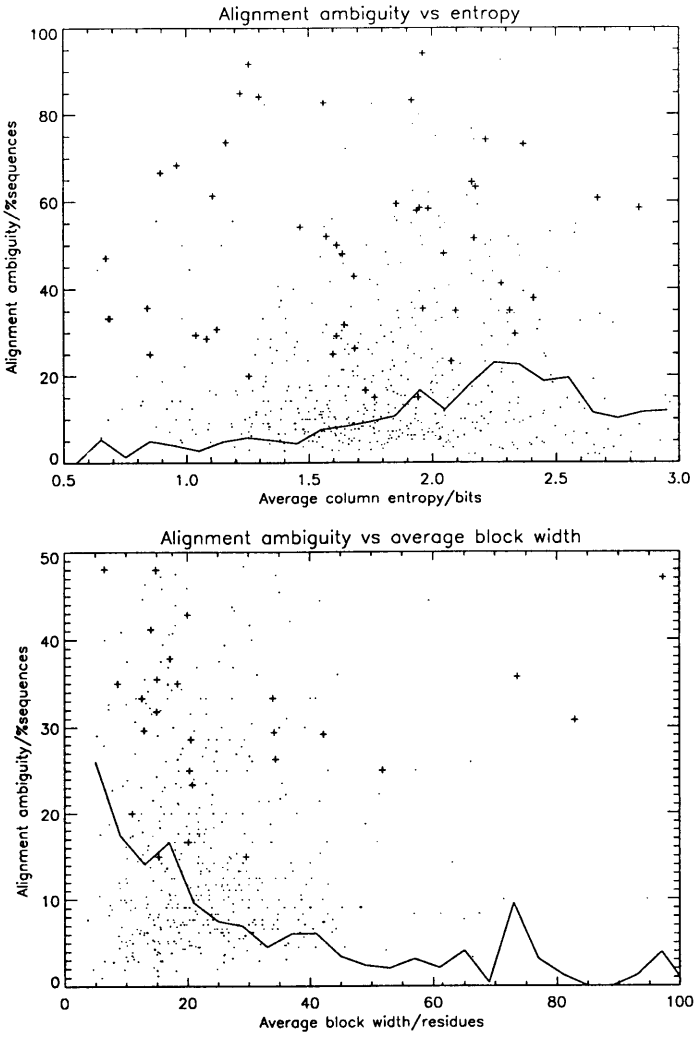

Figure 4.4: These two scatterplots show the proportion of sequences containing ambiguous regions, plotted against the average compositional column entropy (upper plot) and the mean size of ungapped blocks in the alignment (lower plot). Both plots show a distinct correlation. Data points corresponding to alignments with low consensus-accuracy regions are marked with "+" symbols. The solid lines represent y-averages for binned x-values.

4.3 Evaluation: using `postal` as a semi-automated quality check for Pfam

To test-drive the `postal` software, the program was run on each of the 1353 seed alignments in the October 1998 release of Pfam. The seed alignments were then sorted by the number of suspiciously aligned sequences they contained. (A sequence was regarded as suspicious if it had a run of at least 4 residues with probability less than 0.8; these are the default `postal` parameters.)

Of the 1353 seed alignments, 548 have suspiciously aligned regions - a total of 9459 individually suspicious sequences. The fraction of ambiguously aligned sequences is plotted against various properties of the alignment in Figure 4.3 and Figure 4.4. The ambiguity of an alignment appears to be directly correlated to its size (i.e. its length and width - see Figure 4.3). This is intuitively reasonable if poorly-fitting sequence segments are randomly distributed. The average size of ungapped blocks in the alignment and the average column entropy of the alignment, both plausible measures of alignment quality, are also good indicators of ambiguity (Figure 4.4).

As well as calculating site-to-site posterior probabilities for each sequence, `postal` also calculates a reliability indicator for the whole alignment - the "consensus accuracy" - by averaging the probabilities in each alignment column. Like the individual sequence accuracies, the consensus accuracy can be scanned for runs of low values to locate blocks in an alignment where many sequences are ambiguously aligned. In fact, the majority of ambiguously aligned sequences that are detected are due to blocks of this kind, as can be seen from Figure 4.4 where the low-consensus-accuracy regions are marked with "+" symbols. To suppress this effect, `postal` allows masking of low-consensus-accuracy regions; this feature is switched on by default. With low-consensus accuracy masking switched on, the 9459 ambiguously-aligned sequences reduced to 3569, though the number of ambiguous families (510) was comparable to the previous figure (548).

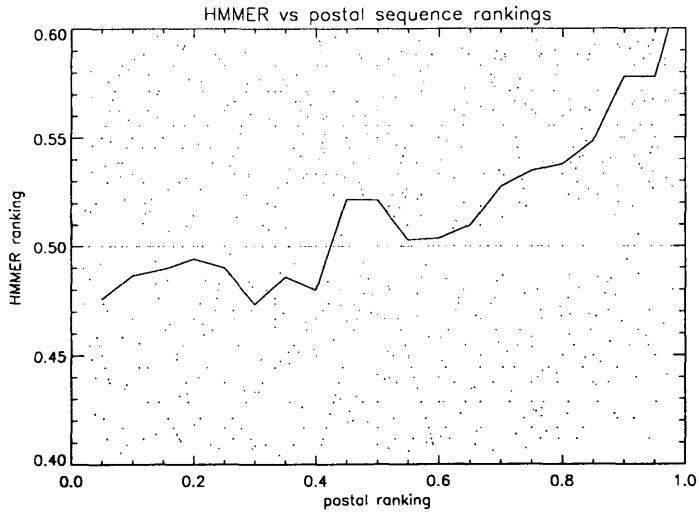

Figure 4.5: The mean rank of a sequence within an alignment, according to HMMER, plotted as a function of the rank according to `postal`. Ranks are fractional, ranging from zero for a poor score to one for a good score. For clarity, only every fifth scatter point is plotted. The solid line represent y-averages for binned x-values.

The worst 20 families in Pfam (without low-consensus accuracy masking) are listed in Table 4.1. Some turn out to be uninteresting from a practical viewpoint, since the low-accuracy stretches correspond to long inserts flanked by weak match states in the HMM and although the alignment of the sequence to the HMM is ambiguous, the effect on the multiple alignment is minor. Other poorly-scoring sequences seem to be outliers, distantly related to the other family members. Since outliers are expected to score poorly in an HMM search anyway, this raises the question: "do posterior probabilities perform any differently at detecting misaligned sequences to straightforward HMMER scores?". It is evident from a plot of the comparative rankings (Figure 4.5) that both methods rank sequences within a particular alignment in a similar way; however, `postal` gives "added value" in that it also reports which parts of an alignment are suspicious.

Accession no.	Family name	Alignment source	%(no.) misfits
PF00065	neur chan	CLUSTALW	94% (48)
PF00128	alpha-amylase	HMM simulated annealing	92% (50)
PF00516	GP120	CLUSTALW	91% (22)
PF00257	dehydrin	CLUSTALW	85% (17)
PF00500	late protein L1	CLUSTALW	84% (16)
PF00125	histone	CLUSTALW	83% (25)
PF00513	late protein L2	CLUSTALW	82% (24)
PF00555	endotoxin	CLUSTALW	82% (19)
PF01298	Lipoprotein 5	CLUSTALW	82% (14)
PF00933	glycosyl hydr14	CLUSTALW	80% (20)
PF00073	rhv	HMM built from alignment	77% (84)
PF00501	AMP-binding	CLUSTALW	76% (23)
PF00067	p450	Structure superposition	75% (48)
PF00009	GTP EFTU	CLUSTALW	74% (46)
PF00089	trypsin	CLUSTALW	74% (46)
PF01010	oxidored q1 C	CLUSTALW	73% (53)
PF00069	pkinase	CLUSTALW	73% (49)
PF00429	ENV polyprotein	CLUSTALW	72% (13)
PF00260	protamine P1	CLUSTALW	68% (13)
PF00360	phytochrome	CLUSTALW	66% (6)

Table 4.1: Top 20 suspicious seed alignments in Pfam. For each family, the fraction of sequences with low-probability runs is indicated, as is the source of the multiple alignment. The high representation of CLUSTALW [THG94a] reflects the fact that 87% of the alignments in Pfam were generated using this program.

4.4 Discussion

The `postal` program provides an indication of the local reliability of multiple alignments by using posterior probabilities as an accuracy measure. Tests on the Pfam database of protein domains lend promise to the program as a practical tool; often, putative low-accuracy regions correspond to areas of the alignment that the alignment algorithm finds ambiguous but that can quickly be resolved by inspection. For a large database like Pfam, manual correction of each alignment is unfeasible and a certain level of automation in the curation is mandatory; a system like `postal` offers a solution to the problem of maintaining the quality of over a thousand multiple alignments.

In Bayesian statistics, the full posterior distribution is generally regarded as a more stable basis for inference than just taking the most likely parameter values. Using posterior probabilities to estimate alignment accuracy is just one example of how this principle could be fruitfully applied to problems in sequence analysis. In common with Lawrence *et al* [ZLL97], it is anticipated that wherever numerical quantities are estimated from alignments, these quantities should be more accurately estimated by averaging over the entire posterior distribution of all alignments, particularly when the sequences are highly divergent and the alignment probability distribution is, consequently, broadly peaked.

4.4.1 Availability

Installation of the `postal` program requires the source code distribution, which includes the HMMER2.0 distribution [Edd95] and is available (under the terms of the GNU public license [GPL]) at the following URL:

`http://www.sanger.ac.uk/Users/ihh/postal/`

Installation and usage instructions are provided on the web site.