Practical RNA Secondary Structure Prediction Using Stochastic Context-Free Grammars

Bjarne Knudsen[1,3] and Jotun Hein[2]

[1]BiRC - Bioinformatics Research Center
University of Aarhus
Høegh Guldbergsgade 10
Building 090
DK-8000 Århus C
e-mail: bk@birc.dk

[2]Department of Statistics
Oxford University
The Peter Medawar Building for Pathogen Research
South Parks Road
Oxford OX1 3SY, England

[3]Corresponding author

May 16, 2002

Key words: RNA secondary structure, stochastic context-free grammar, structure prediction, evolution.

Running head: Practical RNA secondary structure prediction

Abstract

Motivation: RNA secondary structures are important in many biological processes, and efficient structure prediction can give vital directions for experimental investigations. Most available programs for RNA secondary structure prediction only predict structures for a single sequence at a time. This may be sufficient in some applications, but often it is possible to obtain related RNA sequences with conserved secondary structure. These should be included in structural analyses to give improved results.

Results: This work presents a practical way of predicting RNA secondary structure that is especially useful when related sequences can be obtained. It is shown that including these sequences in the analysis greatly improves prediction accuracy.

Availability: RNA secondary structures can be predicted on a web server at http://www.daimi.au.dk/~compbio/rnafold.

Contact bk@birc.dk

Introduction

Many computational methods for RNA structure prediction have been developed. Early algorithms were made by Nussinov *et al.* (1978) and Zuker and Stiegler (1981). Zuker's energy calculations have been further improved (Zuker, 1989; Walter *et al.*, 1994; Mathews *et al.*, 1999), and are probably the most widely used RNA secondary structure prediction method today.

This work improves the basic algorithm of Knudsen and Hein (1999) (here, denoted the $\mathcal{KH}$-99 algorithm), and makes it publicly available.

Algorithm

The algorithm presented in this article is based on the $\mathcal{KH}$-99 algorithm. The original algorithm was only useful for a limited number of sequences, due to its large computation time. For the same reasons, it was not made available to the public. This work makes the algorithm practically useful for larger amounts of sequences. The main concerns are the treatment of gaps, computational speed, and robustness.

The $\mathcal{KH}$-99 algorithm

The $\mathcal{KH}$-99 algorithm uses a stochastic context-free grammar (SCFG) to produce a prior distribution of RNA structures. Given an alignment and a phylogenetic tree relating the aligned sequences, the posterior probabilities of the structures can be calculated. The posterior probability is based on individual probabilities for the columns, or pairs of columns, in the case of a base-pair. Column probabilities are calculated using the likelihood approach by Felsenstein (1981). The evolution of column pairs is modeled using a rate matrix for base-pairs, i.e. a 16 by 16 matrix. The most likely structure is found using the inside-outside algorithm (Lari and Young, 1990). The tree is estimated using a maximum likelihood approach in the SCFG model described.

Gaps

Treating gaps in a sensible way is a returning problem in biological sequence analysis. The best way to deal with gaps would probably be to make an explicit model for insertions and deletions, and use that in the sequence analysis. Unfortunately, such calculations are often complicated, as in statistical alignment (Bishop and Thompson, 1986; Thorne *et al.*, 1991; Hein *et al.*, 2000).

Two simpler ideas are to treat gaps as a fifth nucleotide or to treat them as unknown nucleotides. When using an evolutionary model, many problems arise from treating a gap as a fifth nucleotide. First of all, the frequency of gaps will depend on how many sequences are being analysed and of their evolutionary distance. Furthermore, it cannot be viewed as evolving in the same way as a nucleotide, thus the rates of evolution are difficult to specify.

When treating gaps as unknown nucleotides, the gapped sequence position should have probability one for any nucleotide. This has the advantage that the probability of a column with gaps is equal to the probability of the same column in an alignment without the gapped sequences, and the tree correspondingly pruned, see Figure 1.

From this, it seems that gaps should be treated as unknown nucleotides. This is indeed what is often done in situations where different alignment columns are looked at individually (e.g., Thorne *et al.*, 1996). The problem arises when pairs of columns are analysed together, as is the case for RNA structure prediction. When treating gaps as unknowns, there is no immediate way of controlling whether gaps form pairs with nucleotides, the result of which is shown in the top of Figure 2.

The problem was handled by removing columns where less than 75% of the sequences have nucleotides. The result of this is shown in the bottom of Figure 2. This seems to be a reasonable way to deal with gaps in most cases and was therefore adopted for this algorithm.

Unknown nucleotides

In biological sequences, some nucleotides may be unknown, or only partial information may be available. All these situations can be treated by letting the unknown nucleotide have a probability of one for each of the possible nucleotides. This means, for example, that if a given position is known to be a pyrimidine, its probability of being a U is set to one, and its probability of being a C is also set to one. Using this method, any symbols of the extended nucleotide alphabet can be treated correctly by the algorithm. This is in accordance with Felsenstein (1981).

Tree estimation

In the $\mathcal{KH}$-99 algorithm, the tree was estimated through a maximum likelihood method using the SCFG model. While this gave good results and was interesting with respect to phylogenetic analysis, it was slow. A much faster method is to estimate the tree first. This can be done using standard methods.

In this implementation, pairwise distances between sequences are calculated using maximum likelihood. The rate matrix used should correspond as close as possible to what is used in the $\mathcal{KH}$-99 algorithm. Since the tree is calculated before the structure has been estimated, a single rate matrix has to be used for this purpose. It was made from the $\mathcal{KH}$-99 algorithm by summing the loop rate matrix and a reduction of the base-pair rate matrix to single positions. The rate matrices were weighted with the probabilities that a given position is in a loop region or a stem region, respectively. The tree was calculated from the pairwise distances using the Neighbor Joining algorithm (Saitou and Nei, 1987) and adjusting branch lengths to maximum likelihood estimates. This gave a large increase in speed, since the stochastic context-free grammar calculations only needed to be done once, as opposed to the multiple iterations used in the previous method for estimating the tree.

Robustness

The algorithm assumes that all sequences have exactly the same structure. This means that a single sequence with a slightly different structure might ruin a structure prediction. The same situation applies for alignment errors and sequencing errors. When a single error might change a prediction significantly, the method is not robust. An example is given at the top of Figure 3.

A way to make the algorithm more robust is to let any nucleotide have a small probability of being any other nucleotide. The interpretation of this is most obvious in terms of sequencing errors, but the method works for alignment errors and structure differences, too. Figure 3 shows how introduction of this probability changes the results. In this implementation of the algorithm, the probability was set to 1%.

Partially known structure

It is often the case that something is known about the structure being predicted. Including this knowledge in the analysis can give improved results. Different kinds of knowledge can be included in the analysis:

- That two given columns form a pair together.
- That a given column is involved in a pair.
- That a given column is not involved in a pair.

This can all be included in the calculations by letting the structures that does not satisfy the previous knowledge have a probability of zero. This approach does not change relative prior distributions of allowed structures.

As a side note, no loops of length two is allowed in this implementation, as opposed to the $\mathcal{KH}$-99 algorithm. This was implemented by disallowing pairs between positions of distance less than four.

What structure should be chosen?

A prediction program should of course report a single prediction as the best. The CYK algorithm for finding the most likely parsing from the grammar is often used (Durbin *et al.*, 1998). An alternative to this has been chosen here. The goal of this method is to give the nested structure with the highest expected number of correct predictions. The appendix describes how this structure can be found. Notice that this approach removes some of the problems associated with using CYK with ambiguous grammars.

Once the best nested structure has been chosen, the reliability of the prediction in each position is evaluated relative to the model. Knowing which parts of the prediction are reliable is very important when using the prediction in further work.

Finding a single best structure might not always give all the information that would be useful. To give an overview of the prediction, a dot plot is produced as well. It is a square plot of pairing probabilities for all different pairs. Each probability is represented by the size of a dot in the appropriate position. Probabilities of not pairing are shown on the sides of the plot, see Figure 4. These calculations resemble the work by McCaskill (1990).

Making the obvious individual structure changes

Sometimes, a single sequence will have a slightly longer stem than its relatives. This can be incorporated in the prediction by extending it, if immediate neighbors can form a base pair. Another obvious change is to remove non-standard base pairs from individual sequences. This is done after the structure predictions given by this algorithm.

Implementation

In the implementation of this algorithm, speed and practical usefulness was important.

Calculations

The algorithm has two primary time consuming elements. With an alignment length of n, they are:

- Doing the inside-outside calculations, which has a running time of $O(n^3)$.
- Calculating the column pair probabilities, which has a running time of $O(n^2)$.

It seems that the inside-outside calculations might consume the most time. It turns out, however, that the constant in the column pair probability calculation can be quite large. This part can be the most time consuming of the calculations, when short sequences or many aligned sequences are analysed. The reason is that the column pair probabilities are calculated by multiplying vectors with 16 by 16 matrices in a post-order traversal of the tree. For large trees, many such matrix multiplications has to be done.

The first step in saving time is to avoid doing calculations of column pairs with the same nucleotides more than once. This is effective for trees in the middle size range, because different columns often have the same nucleotides, e.g. all identical. When analysing larger trees, a few mutations have happened in most columns, thus often making them slightly different. To cope with this situation, partial column calculations can be stored for later use.

This has been implemented in the following way: when a calculation is made, the results are stored for the sub-tree below each node. When a new calculation is made, the results up to a given node are re-used if the sequence positions are the same for the sequences in the sub-tree below the node. This drastically reduces the computation time, since no identical matrix multiplications are done twice. It has the cost of a high memory usage and in some cases, a compromise is useful. This was implemented by only saving the results from a sub-tree if it was likely that it would be encountered again. Ideally, this likelihood for nucleotides in a sub-tree should be calculated using the same matrix as in the tree estimation described above. Probabilities for column pairs are closely approximated as the products of the individual columns, since two given columns has a probability of less than $1/n$ of forming a pair and non-pairing columns are treated as independent. Instead of using the matrix from the tree calculations, the matrix for single positions was used, since these calculations are already used in the algorithm.

Output

When a prediction is done, the web-server returns an e-mail to the user. This e-mail contains a link to a web page with the results. On the web page, the following is available in multiple formats:

- A summary of the input.
- The calculated tree.
- The predicted structure.
- Reliabilities of individual predictions.
- A dot plot.

Results

For all predictions in this section, two versions were made. The first version was made using the alignment from the database of the sequences being analysed. This alignment was assumed to be 'correct'. In the second version, sequences from databases were aligned using the CLUSTAL W alignment program by Thompson *et al.* (1994) to imitate a realistic scenario of RNA structure prediction.

Test sets

A number of test sets were made, as described in Table 1. The sets A, B and C are used in evaluating the prediction accuracy as a function of the number of sequences used in the analysis. Test set D is used to show how prediction accuracy varies as a function of evolutionary distance.

Prediction accuracy

An evaluation of the prediction accuracy is shown in Figure 5. The accuracy was calculated as the percentage of positions for which the secondary structure were correctly predicted. For a pairing position to be counted as correct, the position of the predicted pair had to be correct. Sequences in the test sets used here have a maximum pairwise distance of 0.50 units in the Jukes-Cantor model (Jukes and Cantor, 1969). This means that the sequences are quite diverse, but still possible to align without too many errors.

As expected, prediction accuracy rises with the number of sequences used, as more covariance information becomes available. This shows that whenever related sequences are available, they should be used in the structure prediction, if alignable. It also shows that an accuracy of around 75% is obtainable with six sequences. The algorithm can cope with many more sequences, so even higher accuracies could be expected in this case.

Evolutionary distance effect

Two effects of evolutionary distance on prediction accuracy are that large distances implies much covariation information, but it also means that sequences are difficult to align.

These effects are illustrated in Figure 6. When using 'correct' alignments, the accuracy rises with distance, as the evolutionary information increases. The accuracy levels off at around 80%, which seems to be the maximum obtainable average, for two sequences of this type, using this algorithm. The graph made from the CLUSTAL W alignments show how accuracy increases until an evolutionary distance of around 0.6. After this, the accuracy drops due to alignment errors. At a distance of around 0.9, the quality of alignments become so low, that the structures might as well be predicted individually.

Speed

The result of all this is an algorithm that is fast enough to do a structure prediction on a 25 sequence alignment of length 600 in around five minutes on a fast desktop computer. This makes a web server version of the program feasible.

Comparison with other methods

The accuracies obtained by using MFOLD (Zuker *et al.*, 1999; Mathews *et al.*, 1999) with standard parameters are 47.2% for test set A, 68.8% for test set B, 57.0% for test set C and 69.3% for test set D. These results are generally better than for the algorithm described here, using single sequences. When using multiple sequences, accuracies from this algorithm generally become better than for the MFOLD results.

The RNAGA method by Chen *et al.* (2000) also works on a limited set of sequences, rather than single sequences, but has no web-site, on which structures are easily predicted. The method is relatively slow, but seems very promising.

Methods like COVE (Eddy and Durbin, 1994) and Maximum Weighted Matching (Tabaska *et al.*, 1998) depend on more than a few sequences for reliable results (Knudsen and Hein, 1999).

Discussion

Some aspects of this method remain to be explored, as described by Knudsen and Hein (1999). These include: base stacking interactions, a grammar more closely describing real RNA structures and other models for base-pair evolution.

If base-stacking interactions and a better grammar is incorporated in the algorithm described here, the prediction accuracies should become close to the MFOLD results for single sequences, since the methods resemble each other closely, in that situation. For multiple sequences, this algorithm should still be able to perform even better.

As emphasized by this work, an important aspect of RNA structure prediction is the alignment problem. In methods that depend on a sequence alignment, the success of the method is closely linked to the quality of the alignment. Some work has been done in the field of RNA structural sequence alignment (e.g., Gorodkin *et al.*, 1997). The RNAGA method by Chen *et al.* (2000) predicts consensus structures without trying to align the sequences, which might be a useful approach to avoiding the alignment problem.

With this, a new and efficient method of RNA secondary structure prediction is made available via the World Wide Web.

Acknowledgments

I would like to thank Christian N. S. Pedersen for his great help in answering questions about UNIX and C programming. I would also like to thank Ebbe S. Andersen for his help in making this method practically useful. Finally, I thank Jakob S. Pedersen for helpful comments on the manuscript.

Appendix: Finding the best structure

The best structure for the prediction is here defined as the nested structure with the highest expected number of correctly predicted positions. This structure is found by a dynamical programming approach similar to the CYK algorithm, using pairing probabilities which can be calculated from the inside-outside variables. Denote the sub-alignment from position i to j as $C_{i,j}$ (i.e. columns i through j in the alignment) and let C_i represents column i by itself. Write the inside and outside variables as:

$$\begin{aligned} &\text{Inside:} && e_{i,j}(N) = P(N \to C_{i,j}) \\ &\text{Outside:} && f_{i,j}(N) = P(S \to C_{1,i-1} N C_{j+1,n}) \end{aligned}$$

where N is a non-terminal, S is the starting non-terminal, and n is the alignment length. Assume that all pairs are formed by production rules of the type: $N_1 \to dN_2d$, where the d's represent a pair. The probability (under the model) that position i pairs with j is:

$$P_d(i,j) = \sum_{N_1,N_2} f_{i,j}(N_1) P(N_1 \to C_i N_2 C_j) e_{i+1,j-1}(N_2)$$

The sum is effectively only over the rules forming pairs. The probability that position i does not form a pair is:

$$P_s(i) = 1 - \sum_{j \neq i} P_d(i,j)$$

Define $E_{i,j}$ as the maximum possible number of expected correct non-nested predictions in the window from position i to position j in the alignment. Setting $E_{i,j} = 0$, for $i > j$, the $E_{i,j}$'s can be calculated for $i \leq j$:

$$E_{i,j} = \max \begin{cases} E_{i+1,j} + P_s(i) & \text{(Unpaired)} \\ E_{i+1,j-1} + P_d(i,j) & \text{(Paired)} \\ E_{i,j-1} + P_s(j) & \text{(Unpaired)} \\ E_{i,k} + E_{k+1,j} \quad i \leq k < j & \text{(Bifurcation)} \end{cases}$$

The $E_{i,j}$'s can now be calculated in order of increasing $j - i$. Actually, the unpaired part of the recursion could be done by the bifurcation step, by initializing the $E_{i,i}$'s to $P_s(i)$, but the above is easier to interpret. $E_{1,n}$ is the maximum expected number of correctly predicted positions in the alignment. The prediction that gives this maximum can be found by backtracking the above recursion.

Some non-nested structures may exist, which has a higher expected number of correctly predicted positions than the nested one found above. This could happen even though the SCFG used to find pairing probabilities does not take non-nested structures into account. This will be a rare occurrence and would probably not give any practical information. The structure could be found by a maximal weighted matching approach on the pairing probabilities. The Maximal Weighted Matching algorithm is described by Tabaska *et al.* (1998).

References

Bishop,M. J. and Thompson,E. A. (1986) Maximum likelihood alignment of DNA sequences. *J. Mol. Biol.* **190** (2), 159–165.

Chen,J.-H., Len,S.-Y. and Maizel,J. V. (2000) Prediction of common secondary structures of RNAs: a genetic algorithm approach. *Nucleic Acids Res.* **28** (4), 991–999.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological sequence analysis: Probabilistic models of proteins and nucleic acids.* Cambridge: Cambridge University Press.

Eddy,S. R. and Durbin,R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res.* **22** (11), 2079–2088.

Felsenstein,J. (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.* **17** (6), 368–376.

Gorodkin,J., Heyer,L. J. and Stormo,G. D. (1997) Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res.* **25** (18), 3724–3732.

Gorodkin,J., Knudsen,B., Zwieb,C. and Samuelsson,T. (2001) SRPDB (signal recognition particle database). *Nucleic Acids Res.* **29** (1), 169–170.

Hein,J., Wiuf,C., Knudsen,B., Møller,M. and Wibling,G. (2000) Statistical alignment: Computational properties, homology testing and goodnesss-of-fit. *J. Mol. Biol.* **302**, 265–279.

Jukes,T. H. and Cantor,C. R. (1969) Evolution of protein molecules. In Munro,H. N. (ed), *Mammalian Protein Metabolism* pp. 21–123, New York: Academic Press.

Knudsen,B. and Hein,J. (1999) RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics,* **15** (6), 446–454.

Knudsen,B., Wower,J., Zwieb,C. and Gorodkin,J. (2001) tmRDB (tmRNA database). *Nucleic Acids Res.* **29** (1), 171–172.

Lari,K. and Young,S. J. (1990) The estimation of stochastic context-free grammars using the inside-outside algorithm. *Comput. Speech Lang.* **4**, 35–56.

Mathews,D. H., Sabina,J., Zuker,M. and Turner,D. H. (1999) Expanded sequence dependence of thermodynamic parameters improve prediction of RNA secondary structure. *J. Mol. Biol.* **288** (5), 911–940.

McCaskill,J. S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers,* **29**, 1105–1119.

Nussinov,R., Pieczenik,G., Griggs,J. R. and Kleitman,D. J. (1978) Algorithms for loop matchings. *SIAM J. Appl. Math.* **35** (1), 68–82.

Saitou,N. and Nei,M. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* **4** (4), 406–425.

Sprinzl,M., Horn,C., Brown,M., Ioudovitch,A. and Steinberg,S. (1998) Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Res.* **26** (1), 148–153.

Tabaska,J. E., Cary,R. B., Gabow,H. N. and Stormo,G. D. (1998) An RNA folding method capable of identifying pseudoknots and base triples. *Bioinformatics,* **14** (8), 691–699.

Thompson,J. D., Higgins,D. G. and Gibson,T. J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22** (22), 4673–4680.

Thorne,J. L., Goldman,N. and Jones,D. T. (1996) Combining protein evolution and secondary structure. *Mol. Biol. Evol.* **13** (5), 666–673.

Thorne,J. L., Kishino,H. and Felsenstein,J. (1991) An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.* **33** (2), 114–124.

Walter,A. E., Turner,D. H., Kim,J., Lyttle,M. H., Mueller,P., Mathews,D. H. and Zuker,M. (1994) Coaxial stacking of helixes enhances binding of oligoribonucleotides and improves predictions of RNA folding. *Proc. Natl. Acad. Sci. USA,* **91**, 9218–9222.

Zuker,M. (1989) Computer prediction of RNA structure. *Methods in Enzymol-*

ogy, **180**, 262–288.

Zuker,M., Mathews,D. H. and Turner,D. H. (1999) Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide. In Barciszewski,J. and Clark,B. F. C. (eds), *RNA Biochemistry and Biotechnology* pp. 11–43, Dordrecht, the Netherlands: Kluwer Academic Publishers.

Zuker,M. and Stiegler,P. (1981) Optimal computer folding of large RNA sequences using thermodynamic and auxiliary information. *Nucleic Acids Res.* **9** (1), 133–148.

Figure legends

Fig. 1. The effect of treating gaps as unknown nucleotides. Only a single column from the alignment is considered, with the nucleotides put at the leaves of the phylogenetic tree. When gaps are treated as unknown nucleotides the two trees have identical probabilities since leaves with gaps can be removed.

Fig. 2. A structure prediction for three hypothetical sequences. In the top alignment, gaps are treated as unknown nucleotides. The structure, shown as parentheses, include pairs between nucleotides and gaps. In the parenthesis notation, corresponding parentheses indicate positions forming base-pairs. In the bottom alignment, the columns with gaps have been left out of the prediction, because less than 75% of the sequences have nucleotides in these positions.

Fig. 3. In the top alignment, the standard result is given. The bottom alignment shows the structure, when all nucleotides have a 1% chance of being any other nucleotide. The result is a longer stem, which includes one non-standard pair.

Fig. 4. A dot plot made from GCA-tRNA sequences from rat, chicken, mouse and cow (Sprinzl *et al.*, 1998). The lower left corner represents the beginning of the alignment. Imagining the alignment laid out upwards, from here, and toward the right, the dots inside the square represents pairing probabilities between positions. The dots outside the square represents probabilities of not pairing. The tRNA structure turns up quite clearly.

Fig. 5. Accuracy as a function of the number of sequences used in the prediction. Crosses are from results using 'correct' alignments, while boxes are

from CLUSTAL W alignments. Each point represents average results for either all possible combinations of the relevant number of sequences or 50 random combinations, whatever is the lowest number.

Fig. 6. Accuracy as a function of pairwise distances between sequences being analysed. As in Figure 5, crosses are from results using 'correct' alignments, while boxes are from CLUSTAL W alignments. The pairs were grouped according to their Jukes-Cantor distances, in the intervals [0;0.2), [0.2;0.4), [0.4;0.6) etc. The points represent average results for 50 random sequence combinations from a specific range of distances. The x-value of a point is the average of the 50 distances.

Tables

Table 1: Test sets from Gorodkin *et al.* (2001) and Knudsen *et al.* (2001). Sets A, B and C were chosen, so that no pairwise distances within each set is more than 0.5 units in the Jukes-Cantor model (Jukes and Cantor, 1969). Set D was chosen as all unique sequences of length greater than 250 in the eukaryotic SRP RNA database. No two sequences are identical within any of the sets. The average sequence length of each test set is written in parentheses.

Test Set A: 9 tmRNAs (363.8)
act.act., hae.inf., kle.pne., pas.mul., sal.par., sal.typ., she.put., vib.cho., yer.pes.

Test Set B: 13 Bacterial SRP RNAs (270.5)
bac.alc., bac.bre., bac.cer., bac.cir., bac.mac., bac.meg., bac.pol., bac.pum., bac.sph., bac.ste., bac.thu., bre.bre., clo.per.

Test Set C: 10 Eukaryotic SRP RNAs (300.9)
ory.sat., tri.ae-a, tri.ae-b, zea.ma-a, zea.ma-b, zea.ma-c, zea.ma-d, zea.ma-e, zea.ma-f, zea.ma-h

Test Set D: 51 Eukaryotic SRP RNAs (297.4)
ara.th-a, ara.th-b, cae.el-a, cae.el-b, cae.el-c, cae.el-d, can.spe., cin.hyb., dro.mel., fug.rub., hom.sa-a, hom.sa-b, hom.sa-c, hum.ja-a, hum.ja-b, hum.lu-a, hum.lu-b, hum.lu-c, hum.lu-d, lep.col., lyc.es-a, lyc.es-b, lyc.es-c, lyc.es-e, lyc.es-f, lyc.es-g, lyc.es-h, lyc.es-i, lyc.es-j, lyc.es-k, lyc.es-m, lyc.es-n, lyc.es-o, ory.sat., rat.rat., sch.pom., tet.ros., tet.the., tri.ae-a, tri.ae-b, try.br-a, try.br-b, xen.lae., yar.li-a, yar.li-b, zea.ma-a, zea.ma-b, zea.ma-c, zea.ma-d, zea.ma-e, zea.ma-f

Figures

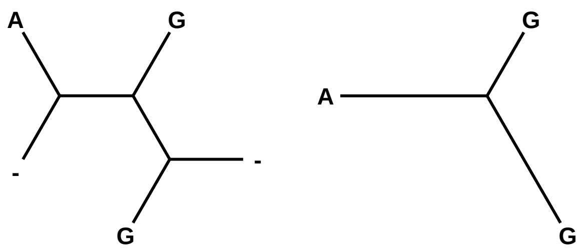


Figure 1:

```
Seq 1      CGAC-----AGCUGAGUGUGACUUUAGAAU
Seq 2      UGACGGUCUAGCUGACUGAUACUUCAGAGU
Seq 3      CGAC-----AGCUGAAUGAGACUUCAGAAU
Structure  ...(((((.....)))))............

Seq 1      CGAC-----AGCUGAGUGUGACUUUAGAAU
Seq 2      UGACGGUCUAGCUGACUGAUACUUCAGAGU
Seq 3      CGAC-----AGCUGAAUGAGACUUCAGAAU
Structure  ...........((((........))))...
```

Figure 2:

```
Seq 1       UGGCG--CUAGCCAUCUGAUACUUCAGAUU
Seq 2       UGACGGACUAGCCAACUGAUACUUCAGAUA
Seq 3       U-ACGUAGUAGCCAUCUGAUACUUCAGAU-
Structure   ...............((((....))))...
```

```
Seq 1       UGGCG--CUAGCCAUCUGAUACUUCAGAUU
Seq 2       UGACGGACUAGCCAACUGAUACUUCAGAUA
Seq 3       U-ACGUAGUAGCCAUCUGAUACUUCAGAU-
Structure   .............((((((....)))))).
```

Figure 3:

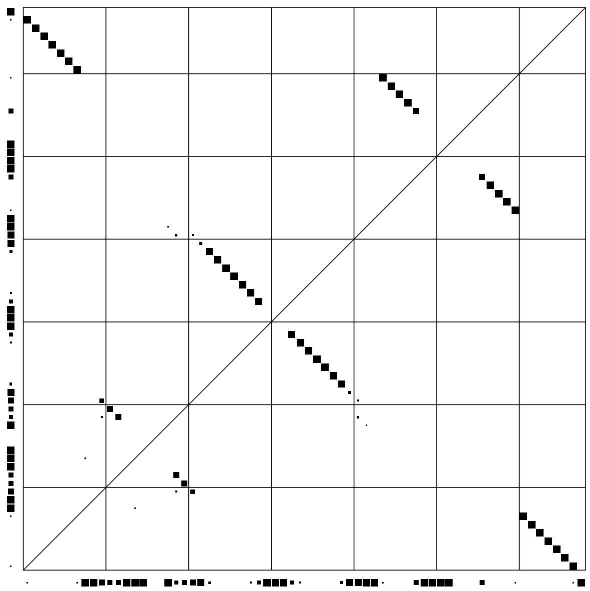
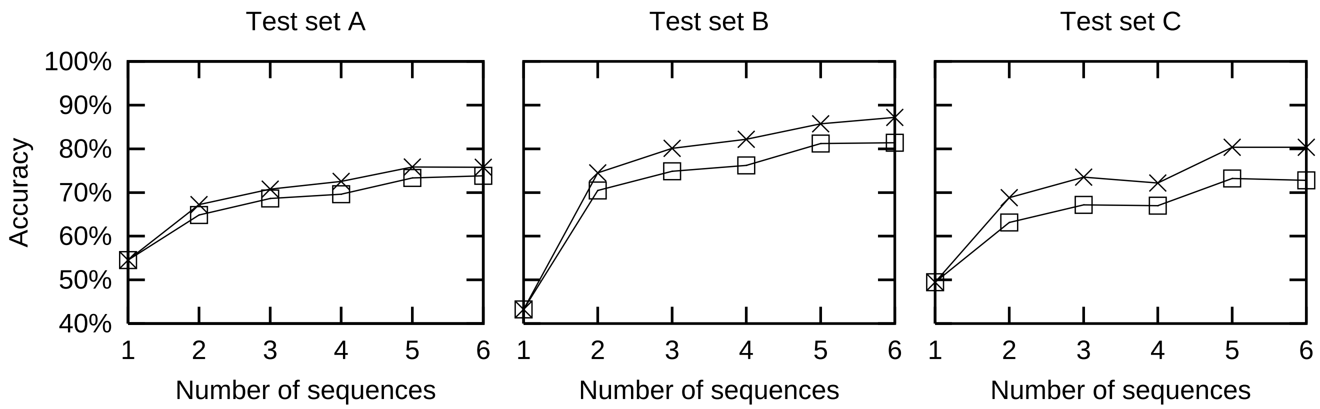


Figure 4:



Test set A
Test set B
Test set C
Accuracy
100%
90%
80%
70%
60%
50%
40%
1 2 3 4 5 6
Number of sequences

Figure 5:

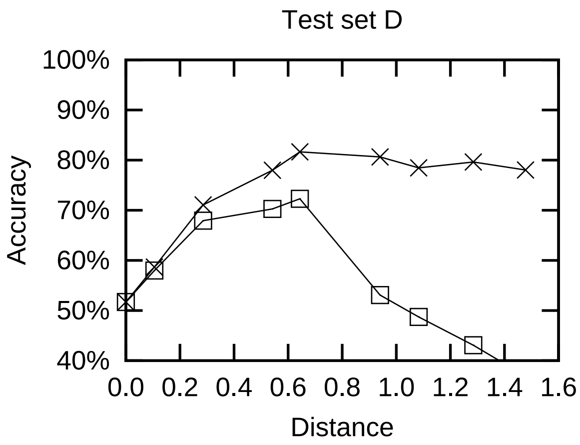

Test set D
100%
90%
80%
70%
60%
50%
40%
Accuracy
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
Distance

Figure 6: