Bioinformatical approaches to RNA structure prediction

&

Sequencing of an ancient human genome

By

Stinus Lindgreen

A dissertation submitted to the University of Copenhagen in partial fulfillment of the requirements for the degree of Ph.D. at the Faculty of Science, University of Copenhagen.

Submitted January 29, 2010

Supervisors:
Professor Anders Krogh
Paul P. Gardner, Ph.D (part 1)
Assistant professor Jakob Skou Pedersen (part 2)
Preface

The work presented in this dissertation sums up the research I have done as a Ph.D student for the past 3 years. During these years, I have been financed by a Faculty stipend from the Faculty of Science at the University of Copenhagen. My initial work was financed by a Novo Scholarship. The work was carried out at the Department of Biology, University of Copenhagen, with a brief stay at the Wellcome Trust Sanger Institute, Hinxton, Cambridge, UK.

I have divided my dissertation into two main parts, reflecting the fact that my research has been focused on two distinct subjects. In the first part, I will give a brief introduction to the field of RNA secondary structure prediction and multiple alignment of structural RNAs. I will also present some first author papers focusing on this subject: My work on measuring basepair conservation in multiple alignments, the development of the algorithm MASTR for simultaneous alignment and structure prediction, and the webserver WAR for performing alignment and structure prediction of RNAs using an ensemble of methods.

In the second part, I will give a short introduction to the field of next generation sequencing and the challenges these new technologies present. I will focus on mapping of reads to a reference genome and subsequent genotyping with focus on ancient DNA. Here, I will also present my first author papers on this subject: The development of the program SNPest for genotyping and SNP calling, and my work on the first ancient genome from a human individual.

The dissertation will end with a conclusion where I will sum up my work and discuss possible future directions of research.

Stinus Lindgreen
January 2010
Contents

Alignment and structure prediction of RNA

Introduction ................................................................. 1
Non-coding RNAs ......................................................... 1
Brief introduction to sequence alignment .............................. 3
Secondary structure prediction of a single RNA sequence .......... 7
Comparative structure prediction ....................................... 13
Simultaneous alignment and structure prediction .................... 16
Combining approaches .................................................. 23

Research Papers

Chapter 1 ................................................................. 31

Chapter 2 ................................................................. 41

Chapter 3 ................................................................. 51

Sequencing and assembly of an ancient genome

Introduction ................................................................. 59
Next generation sequencing .............................................. 59
Ancient DNA ............................................................... 62
Mapping reads to a reference genome ................................ 65
Genotyping ............................................................... 68
First diploid genome of an ancient human individual ............ 74

Research Papers

Chapter 4 ................................................................. 81

Chapter 5 ................................................................. 95

Conclusion
Concluding Remarks ......................................................... 103
Acknowledgements .............................................................. 105
Curriculum Vitae ................................................................. 107
Introduction

In this first part of my dissertation, I will focus on RNA bioinformatics. I will introduce the field of RNA secondary structure prediction and show the connection to RNA multiple alignment. This will lead to a presentation of three first author papers I have in this field, covering different aspects of RNA bioinformatics: Measuring covariation in RNA alignments, simultaneously aligning and predicting the structure of multiple RNA sequences, and combining various RNA structure prediction methods in a webserver.

Non-coding RNAs

In recent years, it has become more and more evident that RNA is of much more importance in living organisms than merely being the mediator of information between the DNA and protein levels. RNA has the ability to both carry genetic information – known from e.g. RNA virus genomes – and act as catalytic agents in the cell as seen in many cases such as Ribosomal RNAs (rRNA) and transfer RNAs (tRNA) important for protein formation, and the RNA parts important in the spliceosome. Many important functional RNAs have been discovered in recent years, making the study of these non-coding RNAs (ncRNA) highly relevant (see e.g. the excellent review in [BFF+05]).

The fact that RNAs are able to both carry genetic information and catalyze biochemical processes has led researchers to propose the RNA World Hypothesis [PJP98, JPP98, Sza99]: The idea that the first life on earth was based on RNA and not protein, since RNA is a simpler polymere to form and can carry out the role of both DNA and protein (although less efficiently). This could also explain why ncRNAs play a key role in many crucial pathways, and why thousands of the human genes do not encode proteins but are transcribed into functional RNA molecules [WHL+05]. Analyzing RNA and being able to predict the secondary structure could therefore present us with key insights into how cells function.

The field of bioinformatical analysis of RNA molecules is therefore growing rapidly. One of the important aspects of ncRNA – as is also the case with proteins – is that the function is intimately tied to the structure of the molecule, thus making methods for RNA structure prediction important. Through evolution, the sequences of related RNAs
can diverge at the nucleotide level while keeping the structure intact. Pure sequence comparison methods therefore fail when applied to ncRNAs. With RNA, you distinguish between primary, secondary and tertiary structure as illustrated in Fig. 1.

The *primary structure* of an RNA is the linear sequence of nucleotides that make up the RNA molecule, normally written in the 5’ to 3’ direction. The four nucleotides adenine, cytosine, guanine, and uracil are abbreviated A, C, G, and U. The pyrimidines (C and U) are the smallest of the nucleotides with a single nitrogen–containing ring, while the purines (A and G) are larger with two nitrogen–containing rings.

The nucleotides can form base pairs by hydrogen bonds. These base pairing interactions constitute the *secondary structure* of an RNA. The base pairs are mainly formed between adenine and uracil and between guanine and cytosine, which are called the *Watson–Crick or classical* base pairs, but non–standard base pairs are seen, especially the wobble base pair between guanine and uracil.

The tertiary structure is formed by contacts between secondary structure elements and constitute the actual three dimensional structure of the molecule. This can include new hydrogen bonds and Van der Waals attractions.

For various reasons, RNA structure prediction focuses on the secondary structure although it is ultimately the tertiary structure that determines the functionality. First of all, the secondary structure forms fast and introduces strong base pairing interactions, thereby contributing the major part of the folding energetics and forming a stable scaffold for the tertiary interactions [OTJ04]. This, combined with the stacking of base pairs into stems, make the secondary structure highly informative. Secondly, the current knowledge about the three dimensional structure is lacking, making it less feasible although some progress is being made [FMT+09].

![Image of RNA structure](image-url)

*Figure 1*: Illustration of primary structure (the sequence at the top), secondary structure (the drawing on the left) and tertiary structure (the model to the right) illustrated using a tRNA molecule. The illustration is from [Gar03].
Brief introduction to sequence alignment

During evolution mutations will change the sequence of nucleotides in the DNA and thus possibly the encoded RNA and protein polymers as well. Therefore, to compare related sequences it is often useful to align them in order to find both segments that are conserved and segments that are variable. The sequence comparison problem is encountered frequently in the biological sciences. Given a number of e.g. RNA or protein sequences, the goal of the alignment is to find similarities as well as differences between them. An alignment is an ordering of the sequences in an $N \times L$ matrix, where $N$ is the number of sequences and $L$ is the length of the alignment. $L$ is at least equal to the length of the longest sequence. Each entry in the alignment matrix contains either a nucleotide/amino acid from one of the sequences or a gap character, normally denoted by ‘-’. A row in the matrix contains a single sequence possibly padded with gaps to have length $L$. A column of the alignment contains $N$ nucleotides/gaps, corresponding to one from each of the aligned sequences.

Sequence alignment is a well studied area in bioinformatics since it is of such general importance. The simple problem of pairwise sequence alignment is well understood and solved to optimality [NW70], whereas the much harder problem of multiple alignment still poses problems – in fact, it has been shown to be $\mathcal{NP}$-hard for a simple cost scheme and an alphabet of size 4 [WJ94]. One of the single most important sequence alignment tools is arguably BLAST [AGM+90,GS93] which is used to find local matches between a sequence and a database – something that most researchers in the biological sciences do routinely.

In the following, all sequences are assumed to be RNA and consist of letters from the alphabet $\Sigma = \{A, C, G, U\}$ containing the four nucleotides, and $\Sigma_G = \{A, C, G, U, -\}$ if the gap character is included. For a natural number $n$, let $\Sigma^n$ denote the set of all words of length $n$ containing only letters from $\Sigma$. The alignment algorithms search for (close to) optimal alignments, defined by either minimizing a cost or maximizing a score. The two approaches are equivalent.

The pairwise alignment problem was solved to optimality using dynamic programming in the well-known Needleman-Wunsch algorithm [NW70]. Using a dynamic programming matrix $M$, the optimal score of a full length alignment of the two sequences is found by aligning longer and longer parts of the two sequences. Each stepwise extension of the alignment is based on the best alignment so far and the score of the next small step: The alignment of the first $i$ nucleotides in sequence $s_1$ and the first $j$ nucleotides in sequence $s_2$ depends on the already calculated alignment of the first $i - 1$ and $j - 1$ nucleotides, respectively, and on either introducing a gap in one of the sequences or aligning two nucleotides. Thus, each step only depends on very simple calculations:

$$M[i, j] = \max\left\{ \begin{array}{l} M[i, j-1] + G \\ M[i-1, j-1] + SC[s_{1}^{i-1}, s_{2}^{j-1}] \\ M[i-1, j] + G \end{array} \right\}$$  \hspace{1cm} (1)
The first and last cases correspond to insertion/deletion events, and the symbol $G$ is the penalty for aligning a nucleotide to a gap. The middle case corresponds to moving parallel to the diagonal in the matrix and aligning a nucleotide from each sequence with each other. The score of this move depends on the scoring scheme used – here denoted $SC$ – where the score for all possible pairings of nucleotides can be found. This $4 \times 4$ scoring matrix can contain specific scores taking into account the physico-chemical properties of the nucleotides or other relevant information. When all entries in the dynamic programming matrix $M$ have been filled, the optimal score is found in the bottom right corner. To find the corresponding optimal alignment – or one possible solution if more than one optimally scoring alignment exist – a backtracking procedure is used to recreate the steps needed to get this optimal score, thus defining the pairwise alignment.

The asymptotic running time of the algorithm is $O(L^2)$ where $L$ is the length of the longest sequence. Since the Needleman-Wunsch algorithm finds the optimal alignment (given the scoring scheme used) between the two full length sequences, the result is a global alignment. In some cases, you might be interested in finding conserved subsequences between the two sequences instead of a full length alignment. This local pairwise alignment can easily be found by a simple extension to the above approach: A fourth option of always choosing $0$ is added to the maximization, thus removing all negative scores and highlighting the positively scoring subalignments. This approach is known as the Smith-Waterman algorithm [SW81] and does not change the asymptotical time complexity of the algorithm.

It should also be noted that the above description assumes a linear gap penalty, where each gap in the alignment is penalized by the same amount $G$. Normally, you would prefer an affine gap penalty, where creating a new gap in a sequence is penalized harder than extending an already existing gap. Also, one should obviously keep information on the steps taken in the alignment process in order to recreate the alignment. Neither of these extensions affect the time complexity.

Although pairwise alignment can be solved optimally using dynamic programming, the problem becomes exceedingly hard as the number of sequences grows. It may seem trivial to go from a pair of sequences to optimizing similarities between $N > 2$ sequences, but it cannot be solved to optimality in polynomial time although a proper formulation of the problem means that an optimal solution does exist. In the general case, multiple alignment is $\mathcal{NP}$-hard [WJ94]. This means that although you can in theory extend the Needleman-Wunsch dynamic programming approach to an arbitrary number of sequences, in practice the focus has been on heuristic methods. Heuristics cannot guarantee optimal solutions but they are often the only practical approach to $\mathcal{NP}$-hard problems where running time and memory usage become huge obstacles.

One of the many difficulties is the fact that it is not trivial to even define the optimality of a multiple alignment [DEKM98]. With pairwise alignment the scoring scheme is straight-forward, and the inherent time-dependent direction in evolution is likewise trivially solved: The relationship between the two sequences does not matter. With
more than two sequences, however, the combinatorics of the possible ways to connect
the sequences grow exponentially, disrupting the clear-cut picture. Ideally, the use of
a phylogenetic tree would sort out the relationships, but this information is not always
available and it is hard to infer such a tree – often this is further complicated by the need
of a good alignment to predict the tree.

A widely used – albeit flawed – way to evaluate a multiple alignment is the *Sum of
Pairs* score, which is a direct generalization of the pairwise alignment scoring scheme
used above. If we assume a linear gap penalty, and we have an alignment $A$ of length $L$
with $N$ sequences, the Sum of Pairs score $SoP$ becomes:

$$SoP(A) = \sum_{i=1}^{L} \sum_{m=1}^{N-1} \sum_{n=m+1}^{N} SC[s_{m}^{i}, s_{n}^{i}]$$

where $SC$ is a $5 \times 5$ scoring matrix defining the substitution score for all possible pairs,
and $s_{m}^{i}$ is the $i$'th letter (nucleotide or gap) in sequence $m$ in the alignment. From the
sum we see that $\frac{N(N-1)}{2}$ pairs are included, which is in the order of $O(N^2)$. This is
problematic from an evolutionary point of view since mutations are overcounted. In re-
ality, each of the $N$ sequences has only one direct ancestor, and the number of necessary
pairwise alignments is therefore in the order of $O(N)$.

To get around the problems with $SoP$, different alternatives have been proposed: The
TKF91- and TKF92-models [TKF91, TKF92] present a probabilistic framework where
evolutionary constraints are incorporated by assigning probabilities to substitutions and
indels. Although a probabilistic scoring scheme does not in itself guarantee the correct
behaviour – the model could still be flawed in many ways – the result is often easier
to understand, and comparing results becomes more intuitive. Furthermore, the TKF-
models have the advantage that they can be cast as Hidden Markov Models, making
it possible to apply the well–known Viterbi [Vit67] and forward/backward algorithms
[Rab89].

In [DEKM98], it is suggested to use the entropy of the alignment as a cost func-
tion and minimize this value by increasing the similarity of the alignment columns. As
opposed to Sum of Pairs, this approach considers the columns as a whole and avoids
the problem of overcounting individual mutations. A simplification is introduced by as-
suming site independence in the calculations – this is obviously not true. However, one
could change the alphabet by increasing the word size and consider e.g. dinucleotides or
codons. This would introduce small range dependencies, and conditional entropy could
be used as the cost function instead [CT91].

For the sake of being general, assume a word size $n \geq 1$, where for $n = 1$ the
following expressions reduce to the standard single nucleotide entropy [DEKM98]. A
word $w \in \Sigma^{n}$ can be broken into a (possibly empty) prefix $w' \in \Sigma^{n-1}$, containing
the first $n - 1$ letters of $w$, and the final character $a \in \Sigma$ – i.e $w = w'a$. Let $c_i(w)$
be the number of occurrences of word $w$ ending at position $i$ in the alignment. Given
the decomposition of \( w \), this can be written as \( c_i(w' a) \) which is simply the count of character \( a \) at position \( i \) preceded by \( w' \). The conditional probability of observing word \( w \) at position \( i \) can be written as:

\[
P_i(w) = P_i(a|w') = \frac{c_i(w' a)}{\sum_{a' \in \Sigma} c_i(w' a')},
\]

where the summation gives the count of all words starting with the same prefix \( w' \) and ending at position \( i \). If the word length is \( n = 1 \), this is just the fraction of nucleotide \( a \) at position \( i \).

Given the set of all words of length \( n \) ending at position \( i \), \( w_i \in \Sigma^n \), the probability of the entire multiple alignment \( MA \) is given as:

\[
P(MA) = \prod_{i=1}^{L} \prod_{w_i \in \Sigma^n} P_i(a|w')^{c_i(w' a)}
\]

To get the conditional entropy \( H \), we take the negative logarithm of the probability:

\[
H(MA) = -\sum_{i=1}^{L} \sum_{w_i \in \Sigma^n} c_i(w' a) \log P_i(a|w')
\]

The second sum can be taken over all possible words \( w \in \Sigma^n \) instead of just words ending at position \( i \), since the counts \( c_i(w) \) of words not occurring at position \( i \) are 0 and thus will not affect the entropy. The conditional entropy should be normalized with respect to the mean of the ungapped sequence lengths \( \bar{L} \) and the number of sequences \( N \) to yield a cost function:

\[
\text{cost}_H(MA) = -\frac{1}{\bar{L}} \sum_{i=1}^{L} \frac{1}{N} \sum_{w \in \Sigma^n} c_i(w' a) \log \left( \frac{c_i(w' a)}{\sum_{a' \in \Sigma} c_i(w' a')} \right)
\]

In case the sequences contained only nucleotides, the conditional entropy would be the final cost function. However, since we are dealing with multiple alignments, the alphabet contains the gap character which will not be penalized in this framework. Minimizing the entropy will only lead to gaps being aligned in columns (to maximize the similarity) but not affect the number of gaps. A separate gap penalty will therefore be necessary although that will compromise the clean probabilistic interpretation of the cost. However, a purely probabilistic formulation including gaps can be obtained using HMM formalism, which will be shown in the description of the MASTR algorithm.

One thing is defining a scoring scheme for the multiple alignment, another to actually perform the alignment of the sequences. Many approaches have been suggested, but one of the most successful strategies is the progressive alignment method described by
Feng and Doolittle [FD87] and popularized in ClustalW [THG94]. In this approach, a
guide tree is used to find the most similar pair of sequences. This closest pair is aligned
using Needleman-Wunsch and (in the case of ClustalW) a profile is generated which
replaces the two sequences, thus making the problem one sequence smaller. A profile is
a representation of an alignment as a weight matrix of size $4 \times L$, giving the distribution
of letters at each position. The Needleman-Wunsch algorithm can be extended to handle
profiles and sequences [DEKM98]. In the next step, the new closest pair – perhaps
containing a profile – is found, a new pairwise alignment is performed, and the process
is iterated until all sequences have been aligned.

The major drawback with progressive alignment is that early errors in the process are
inherited all the way through. At each step, by necessity only a local view of the problem
is taken. The bottom-up way of aligning the sequences does not make it possible to fix
errors that become evident at a later stage. Other methods such as MAFFT [KMKM02,
KKTM05] and T-Coffee [NHH00] try to solve this problem by re-aligning the sequences
and performing different types of consistency checks on the pairwise alignments.

**Secondary structure prediction of a single RNA sequence**

As mentioned previously, the secondary structure of RNAs is of great importance as they
contribute a major part of the folding energetics and hence the stability of the 3D struc-
ture. Since structure is tied to function, the prediction of potential structure is relevant
in many aspects.

There are certain basic principles that are assumed in the prediction of secondary
structure: A base pair between the $i$'th and $j$'th nucleotide in a sequence is written
as $(i,j)$. Assume without loss of generality that $i < j$. Due to constrictions on the
flexibility of the backbone chain, there must be at least three nucleotides between two
positions before a base pair can form, i.e. $j - i > 3$. Also, a nucleotide can only base
pair with at most one other nucleotide.

Assume we have two base pairs, $(i,j)$ and $(k,l)$, where $i < k$. The base pairs can
be positioned in three ways relative to each other: Sequentially where $i < j < k < l$,
embedded where $i < k < l < j$, or crossing where $i < k < j < l$. The third case
is called a **pseudoknot** as illustrated in Fig. 2. Pseudoknotted structures are known to
occur and be of functional importance, but they are very hard to predict computationally.
Therefore, most methods do not consider these (some exceptions are [RE00, RG04])
and pseudoknots will be ignored in the following. Pseudoknotted structures could also
arguably be defined as tertiary interactions as they rely on the formation of base pairs
between secondary structure elements.

In the cell, the formation of secondary structure interactions happens 100,000 times
faster than unfolding [OTJ04] due to the stacking of base pairs or nearest–neighbour
interactions [BDTU74, OTJ04]. A base pair in isolation is energetically unfavorable,
whereas two or more base pairs in a row are highly favorable, making the formation of
Figure 2: Illustration of two simple pseudoknotted structures. **Left:** Two hairpin loops are formed (blue base pairs), and the tertiary structure brings the unpaired loop regions together, making additional (red) base pairs possible. **Right:** A hairpin loop is formed (blue base pairs), and the RNA molecule folds back and forms additional (red) base pairs with the unpaired loop region.

base pair stems the single most stabilizing feature in RNAs [BDTU74]. The unpaired loop regions are energetically unfavorable, although experiments have shown that some sequences in loops are more favorable than others [ATJ92]. This setup of energetically favorable stems on the one hand and unfavorable loop regions on the other lead to a large solution space of possible RNA structures for a single sequence.

One of the first methods to predict the structure of an RNA sequence was the Nussinov algorithm [NPGK78]. In this approach, the goal is to maximize the number of base pairs in the structure and hence the stability of the folded RNA. This dynamic programming procedure has many similarities to the Needleman-Wunsch algorithm for pairwise alignment previously described in Eq. 1. The problem with the size of the solution space is solved by basing the calculations on the optimal folding of subsequences of the RNA sequence \( s \). A dynamic programming matrix \( S \) is used to find the optimal structure of longer and longer subsequences of \( s \). As opposed to Needleman-Wunsch, in this case the sequence is compared to itself in order to find internal base pairing interactions.

In the entry \( S[i,j] \), the optimal score is recorded for folding a subsequence \( s_{i,j} \) going from nucleotide \( i \) to nucleotide \( j \) in \( s \). Just as with pairwise alignment, the possibilities can be condensed into a few simple calculations – in this case based on whether nucleotides \( i \) and \( j \) form a base pair or not. The simplest \( O(L^2) \) time formulation of the Nussinov algorithm only finds stems:

\[
S[i,j] = \max \begin{cases} 
S[i+1,j] \\
S[i,j-1] \\
S[i+1,j-1] + \beta
\end{cases}
\]  

If we do not add a new base pair to the structure, the optimal score of the subsequence is the same as for the either of the subsequences that is one nucleotide shorter (the two first cases). If a base pair \((i,j)\) is added, the new optimal score is the sum of the score of subsequence \( s_{i+1,j-1} \) and the base pair parameter \( \beta \). The upper triangular matrix \( S \) is initialized by observing the fact that some subsequences cannot form base pairs: \( S[i,j] = 0 \), for \( j - i < 4 \). In the end, the optimal score is found in entry \( S[1,L] \).

The base pair parameter \( \beta \) can be used to decide what the recursion maximizes. In
the standard case, $\beta = 1$ for allowed base pairs leading to the structure with maximal number of base pairs. One could also incorporate a very simple energy model in the algorithm by letting $\beta$ depend on the base pair in question to e.g. prefer C-G base pairs.

To allow the algorithm to detect branching structures, a fourth case has to be added to the recursion:

$$\max_{i < k < j} (S[i, k] + S[k + 1, j])$$

This case locates the nucleotide $k$ between $i$ and $j$ that maximizes the total number of base pairs by considering the two subsequences $s_{i,k}$ an $s_{k+1,j}$ separately. The extra calculations needed for looking through the $O(L)$ possibilities increase the time complexity to $O(L^3)$.

Considering the structure prediction problem simply as a matter of maximizing the number of base pairs is too simplistic. As mentioned previously, interactions such as base pair stacking largely determines the free energy of the structure meaning that the single base pairs cannot be considered in isolation. Instead, one should consider the physico-chemical properties of RNAs and model the energetics of the folding process. This approach seeks to minimize the free energy of the structure and thus find the most stable structure.

This minimum free energy (MFE) scheme was formalized by Zuker and Stiegler [ZS81]. Experimentally, the energy contributions from the various parts of small RNA molecules are obtained [MSZT99]. These values are then used in calculations that seek to minimize the MFE of large, real life RNAs. The assumption is that the energy of the complete structure can be approximated by summing over the individual structural parts of the RNA [TJBD+73]. Some subtle interactions will necessarily be missed by considering the structure in parts, but the overall performance outweighs this loss.

The mfold algorithm [ZS81] is the original MFE algorithm that will be described in the following, but RNAfold from the Vienna package [HFBS94] is also a widely used MFE folding program for single RNA sequences. The procedure used in mfold can be conveniently defined in terms of graph theory by viewing the structural parts of an RNA as so-called $k$-loops. However, since this is not directly used in my work I will not go into the details here but refer instead to my Master thesis [Lin05] where I describe $k$-loops in some detail. In the following, mfold is described in more general terms.

This algorithm, too, uses dynamic programming although it is slightly more complex than the two previous examples. The dynamic programming matrix $W$ contains the MFE for subsequences of $s$, and the dynamic programming matrix $V$ contains the MFE given that the two outermost nucleotides in the subsequence form a base pair. Let $E_{HL}(i,j)$ denote the free energy contribution from the hairpin loop closed by the base pair $(i,j)$. Let $E_S(i,j,i',j')$ denote the free energy contribution from one of the structural elements in stems (i.e. stacking of two base pairs, a bulge on either side of the stem, or an internal loop) closed by the base pairs $(i,j)$ and $(i',j')$. Let $E_k(u,k-1)$ denote the energy contribution from a $k$-loop, $k > 2$. The $k$-loop consists of $k$ base pairs, of which $(i,j)$ is
one, and contains \( u \) unpaired nucleotides. The MFE structure for an RNA sequence can be found using the recursion:

\[
W(i, j) = \min \begin{cases} 
W(i + 1, j) & \text{\( i \) dangling end} \\
W(i, j - 1) & \text{\( j \) dangling end} \\
V(i, j) & \text{\((i, j)\) base pair} \\
\min_{i < i' < j - 1} (V(i, i') + W(i' + 1, j)) & \text{open bifurcation}
\end{cases}
\]

\[
V(i, j) = \min \begin{cases} 
E_{HL}(i, j) & \text{hairpin loop} \\
E_S(i, j, i', j') + V(i', j') & \text{stem element} \\
\min_{i + 1 < i' < j - 2} (E_k(u, k - 1) + W(i + 1, i') + W(i' + 1, j - 1)) & \text{\( k \)-loop}
\end{cases}
\]

When calculating matrix \( W \), the cases mirror the ones seen in the Nussinov recursions, Eqs. 3 and 4. The first two cases have an unpaired nucleotide in either the \( i \) or \( j \) side, and the MFE is found by looking at subsequences that are one nucleotide shorter. If \( (i, j) \) form a base pair, the MFE is found in the \( V \) matrix as defined above. If we have a \( k \)-loop, the structure is divided into two parts, and the MFE is found as the sum of the two subsequences. In that case, a base pairing partner \( i' \) for nucleotide \( i \) is found and the sequence is split into \( s_{i,i'} \) and \( s_{i'+1,j} \). The first subsequence has a base pair between \( i \) and \( i' \), and therefore the \( V \) matrix is used for this part, while that is not necessarily the case for the latter subsequence, and hence the \( W \) matrix is used.

For the \( V \) matrix, the cases depend on the type of structural element we have encountered: Hairpin loops are entirely defined by the loop of unpaired nucleotides and the closing base pair \( (i, j) \). The energy contribution from structural elements in stems is given as the contribution from the element itself, i.e. \( E_S \), and the energy of the structure following the closing base pair \( (i', j') \), i.e. \( V \). For \( k \)-loops, the energy contribution is based on the loop asymmetry where the number of unpaired nucleotides between the base pairs constituting the loop determines the energy.

The original mfold algorithm has been improved over the years: In 1999, Lyngsø et al. [LZP99] improved the time complexity to \( \mathcal{O}(L^3) \), mainly by analyzing how the structural parts of stems were treated. In [MT02], the predictions were improved by refining the treatment of \( k \)-loops in the recursion, which were originally dealt with using a linear approximation.

The algorithm guarantees that the structure found has minimal free energy as determined by the energy parameters. However, the biologically active form might not be at the MFE but actually have a higher free energy as illustrated in Fig. 3. Also, simplifications – however necessary they might be – are inherent in the programs (e.g. the energy parameters are based on small RNAs and extrapolated to full-size structures, and the splitting of a structure into its constituent parts is a simplification) meaning that the prediction might not be at the actual energy minimum. Finally it is known that the
RNA molecules are not static structures but flexible and dynamic [OTJ04]. This means that the search for one structure is, in a sense, the wrong path. A solution can be to report not just the MFE structure but also some ensemble of structures close to the minimum [Zuk89, WFHS99].

Figure 3: Illustration of the caveats using MFE using a tRNA from *S. Cerevisiae* as an example. The red graph shows the free energy of the structure space. The MFE structure is found at the far left, showing a non-functional conformation of the RNA molecule. The biologically active form is found to the right with a higher free energy and this will therefore not be predicted by MFE methods. The figure is generated by Paul P. Gardner and copied from http://en.wikipedia.org/wiki/File:Yeast.png

A slightly different approach to structure prediction is to consider the Boltzmann ensemble of possible RNA structures for a given sequence. This can be seen as a probability distribution over structure space $S$, where the probability for a given structure $S \in S$ is determined by the free energy of the structure, $E(S)$. This leads to the calculation of the partition function $Q$ which is the sum over all states – i.e. structures – in the solution space. For RNA, the calculation of the partition function was described by McCaskill in 1990 [McC90].

Let $S$ be the set of all possible structures for the RNA sequence, and let $E(S)$ be the free energy of the structure $S$. For a given temperature $T$, the partition function is given as:

$$Q = \sum_{S \in S} e^{-\frac{E(S)}{k_B T}}$$

where $k_B$ is the Boltzmann constant. The lower the free energy $E(S)$ of a structure
is, the more weight this structure has in the summation. Since the number of possible structures for a sequence grows exponentially in the sequence length $L$ [Wat95], a direct enumeration of all possible structures is intractable. Instead, a $O(L^3)$ dynamic programming approach is used.

Mirroring the expressions for mfold given above, let $Q_{i,j}$ be the partition function calculated for the subsequence $s_{i,j}$, and let $Q^{BP}_{i,j}$ be the partition function on the same subsequence under the constraint that the nucleotides $i$ and $j$ form a base pair. Thus, the partition function is given as $Q = Q_{1,L}$. The recursion is defined over increasing subsequences of $s$, and for subsequence $s_{i,j}$ the partition function with the base pairing constraint is given as:

$$Q^{BP}_{i,j} = e^{-\frac{E(i,j)}{k_B T}} Q_{i+1,j-1}$$

$E(i,j)$ is the free energy contribution from the new base pair $(i,j)$, which is combined with the general partition function already calculated for the shorter subsequence $s_{i+1,j-1}$ without restriction on base pairing.

The unrestricted partition function on the subsequence $s_{i,j}$ is calculated as:

$$Q_{i,j} = Q_{i+1,j} + \sum_{k=i+1}^{j} Q^{BP}_{i,k} Q_{k+1,j}$$

The first term is the unrestricted partition function on the subsequence without nucleotide $i$. The second term is the sum over all possible partitions of the subsequence $s_{i,j}$ where a base pair is formed between nucleotide $i$ and some nucleotide $k$ in the remaining sequence. Thus, all base pair combinations are included in the sum, ultimately yielding the partition function defined over the entire structure space $S$.

Having calculated $Q$, the probability of a specific structure $S$ is easily found:

$$P(S) = \frac{e^{-\frac{E(S)}{k_B T}}}{Q}$$

The lower the free energy, the higher the weight of the structure is in the sum, and the more likely that structure becomes as a candidate for the native structure. The information in the partition function can also be used to calculate the probability $P(i,j)$ of observing a specific base pair $(i,j)$ in any of the possible structures a given RNA can adopt. Let $S_{i,j}$ be the set of all structures containing the base pair $(i,j)$:

$$P(i,j) = \sum_{S \in S_{i,j}} P(S)$$

The sum is over all structures containing the base pair and thus we get the probability of that base pair occurring in the Boltzmann ensemble. The base pair probabilities $P(i,j)$ can be used to e.g. define a scoring function for base pairs by using the probabilities directly in the prediction. The calculation of the partition function is included in the Vienna package [HFBS94].
Comparative structure prediction

In the above, the focus was on predicting the structure – or an ensemble of structures – of a single RNA sequence. A more fruitful approach to secondary structure prediction is to utilize the information inherent in multiple alignments of a set of related RNA sequences. For functional RNAs, the structure is often more conserved than the sequence. Given that the alignment is correct, it will be possible to locate mutations in the nucleotide sequences that preserve the base pairing potential. This can be seen as strong evidence for structure conservation and hence for important structural features. In the following, I will first look at methods to predict secondary structure given a multiple alignment of RNA sequences, followed by methods for simultaneous alignment and structure prediction of RNAs.

One would expect a set of related RNA sequences to share a common structure. Given an alignment of the sequences, the base pairing columns in the alignment can be inferred by covarying positions, i.e. positions where a mutation in column $i$ (relative to the consensus nucleotide at that position) is matched by a corresponding mutation in column $j$ that preserves the base pair. For instance, if the sequences have a common base pair $(i, j) = A \cdot U$, a mutation in a sequence at position $i$ that changes $A$ to $C$, could be matched by a mutation at position $j$ that changes $U$ to $G$, thus creating a $C \cdot G$ base pair keeping the structure intact. This is an example of a compensatory mutation.

Given an alignment, a measure of covariation is needed to find these base pairing columns, and the goal is to find column pairs that show a higher degree of covariation than expected by chance. The classical measure is mutual information content [Sha48], $MI_{i,j}$, which in the context of RNA alignments is defined on pairs of columns, $i$ and $j$. Let $f_i(a)$ be the frequency with which nucleotide $a$ is observed in column $i$. Let $f_{i,j}(a,b)$ be the frequency with which we observe nucleotide $a$ in column $i$ and nucleotide $b$ in column $j$. We then compare the actual frequency of nucleotide pair $(a, b)$ with the expected frequency given by the frequencies of $a$ and $b$ for all pairs of nucleotides:

$$MI_{i,j} = \sum_{a,b \in \Sigma^2} f_{i,j}(a,b) \log \frac{f_{i,j}(a,b)}{f_i(a)f_j(b)}$$

This is identical to the Kullback-Leibler divergence of the product of the two marginal distributions $f_i(a)$ and $f_j(b)$ from the joint distribution $f_{i,j}(a, b)$ [KL51]. The mutual information content for a column pair $(i, j)$ tells you how much information is gained about column $i$ if you are told what column $j$ looks like and vice versa (the score is symmetric). The higher the mutual information content, the higher the information gain. The maximum mutual information is $\log n$, where $n$ is the alphabet size. In the case of RNA alignment, the maximum is thus $\log(4)$.

There are some problems with mutual information. If the column pair is highly conserved, the mutual information drops towards zero. This means that a completely
conserved base pair in the alignment will not be detected by this measure. For detecting base pairs, MI does not discriminate between structurally important pairs and column pairs that cannot form base pairs but show some degree of covariation nevertheless. Also, it is not clear how to deal with gaps in the alignment.

These shortcomings have led to the development of alternative measures of covariation. In the program RNAalifold [HFS02] the common secondary structure for a multiple alignment of RNA is predicted based on a combination of covariation and folding energy. These two entities are combined in a cost function $\beta_{i,j}$ over the alignment columns. RNAalifold takes an alignment as input containing $N$ sequences $s_a, a = 1, 2, \ldots, N$. Since they are aligned, all sequences have the same length $L$. For each sequence $s_a$, a base pairing matrix $\Pi^a$ indicates the possible base pairs in the sequence: If base pair $(i,j)$ can form, then $\Pi^a_{i,j} = 1$ or else $\Pi^a_{i,j} = 0$. The optimal structure $S^*$ is the set of base pairs minimizing the sum of $\beta_{i,j}$.

The optimal structure is found by recursively solving the problem for subalignments $A_{i,j}$ containing columns $i$ through $j$. Let $E_{i,j}$ be the optimal cost of a structure spanning a given subalignment. The optimal cost of the full alignment, $E_{1,L}$, is then found by dynamic programming:

$$E_{i,j} = \min \left\{ E_{i,j-1} \min_{\Pi_{i,k} = 1} \left[ k = i+4, \ldots, j \right] (E_{i+1,k-1} + E_{k+1,j} + \beta_{i,k}) \right\}$$

The recursion exploits the optimal substructure of the problem: If no new base pair is formed, the optimal cost is found as the cost of the subalignment that is one column shorter. If a new base pair is formed between column $i$ and some column $k < j$, the problem is split into two. One subalignment covers the columns $i + 1$ to $k - 1$, and the other covers columns $k + 1$ to $j$. The optimal position $k$ to split the alignment is found by scanning through the $O(N)$ possible choices, and the cost of the new base pair $\beta_{i,k}$ is added to the optimal costs of the two subalignments. The triangular matrix $E$ contains on the order of $O(N^2)$ entries, giving a time complexity of $O(N^3)$.

The cost function $\beta$ combines an energy term with a novel covariation measure that deals with the problems presented by standard mutual information. Let $x^a_i$ be the character at position $i$ in sequence $a$. A Hamming-like distance measure between two sequences for a given column pair is introduced:

$$d^{a,b}_{i,j} = 2 - \delta(x^a_i, x^b_i) - \delta(x^a_j, x^b_j)$$

The $\delta$-function is 1 if the two characters are identical and 0 otherwise. The distance $d$ between two sequences is then 0 if they are identical at both positions, 1 if they differ in one column, and 2 if they differ in both positions. This distance measure is then used to calculate a covariation score $C_{i,j}$ as the mean distance $d_{i,j}$ between all $\binom{N}{2}$ pairs of
sequences in the alignment that can form a base pair:

\[ C_{i,j} = \frac{1}{N(N-1)} \sum_{a=1}^{N} \sum_{b=a+1}^{N} d_{i,j}^{a,b} \Pi_{i,j}^{a} \Pi_{i,j}^{b} \]

A penalty term is introduced to account for sequences that cannot form a base pair where other sequences in the alignment can:

\[ q_{i,j} = 1 - \frac{1}{N} \sum_{a=1}^{N} (\Pi_{i,j}^{a} + \delta(x_{i}^{a}, -) \delta(x_{j}^{a}, -)) \]

The summation counts the number of base pairing sequences (where \( \Pi_{i,j}^{a} = 1 \) and both \( \delta \)-terms are 0) and the number of gap-pairs (where \( \Pi_{i,j}^{a} = 0 \) and the two \( \delta \)-terms are both 1) since both are considered consistent – i.e. the pair is either conserved (the first case) or both base pairing partners are deleted (the second case). Sequences containing a single gap (where \( \Pi_{i,j}^{a} = 0 \) and one \( \delta \)-term is 1 and the other 0) or sequences with two nucleotides that cannot form a base pair (where \( \Pi_{i,j}^{a} = 0 \) and both \( \delta \)-terms are 0) contribute 0 to the sum. The \( q_{i,j} \) measure is therefore the fraction of sequences inconsistent with a base pair between columns \( i \) and \( j \).

The evidence for a base pair in the alignment is a weighted sum of the covariation and the penalty term:

\[ B_{i,j} = C_{i,j} - \phi_{1} q_{i,j} \]

The \( \beta \)-cost function combines the alignment evidence above with the energy terms from [MSZT99]. Let \( \epsilon(x_{i}^{a}, x_{j}^{a}) \) be the energy contribution from forming a base pair between nucleotides \( i \) and \( j \) in sequence \( s_{a} \). The energy contribution from having a base pair between columns \( i \) and \( j \) is then the average energy from all sequences, \( \epsilon_{i,j} \), giving the cost function as a weighted sum of the two terms:

\[ \beta_{i,j} = \epsilon_{i,j} - \phi_{2} B_{i,j} \]

There are many other methods that predict the consensus structure given an alignment. Some notable examples are Pfold [KH99, KH03] using stochastic context free grammars (SCFG) [DEKM98] to find the most probable structure given an alignment and a phylogenetic tree, RNASampler [XJS07] that combines base pair probabilities with nucleotide alignment probabilities to sample possible stems, and R-Coffee [WHN08] that extends the T-Coffee program [NHH00] to find alignments consistent with the predicted secondary structures of the sequences.

Given that many methods try to solve the secondary structure prediction problem using some measure of covariation extracted from a multiple alignment, it is relevant to evaluate the predictive power of various measures. If a specific measure is shown to be a good predictor of base pairing interactions, that result will be of importance to researchers working with comparative structure prediction. Likewise, if a measure turns out to be a bad predictor, that too is important. Finally, since the signal from the
alignment depends on the sequence identity, it might also be the case that one measure is better suited for sequences of low identity, while another works better for more similar sequences.

In the paper “Measuring covariation in RNA alignments: Physical realism improves information measures” presented in chapter 1, we analyze various measures of covariation. The work was done by assembling sets of RNA sequence alignments of varying overall identity with annotated reference structures from Rfam [GDT+08]. We evaluate eight different measures of covariation – of which three were previously described in the literature and the remaining five were novel measures – by testing their ability to discriminate between the true (annotated) base pairing columns and other column pairs. In this paper, the best performing measure was shown to be the covariation term from RNAalifold expanded to include stacking information.

Simultaneous alignment and structure prediction

As described, various methods use the information in sequence alignments when predicting the structure of multiple related RNA sequences. The shortcoming in all these approaches is the need for a high quality alignment. If the alignment is wrong, the predicted consensus structure will by necessity also be at least partially wrong. In order to make the correct alignment, the conserved structure should be taken into account, and the conserved structure should be predicted based on the correct alignment. To avoid this circular dependency, some methods attempt to solve both problems simultaneously.

The original exact algorithm for this problem was described by Sankoff in 1985 [San85]. The algorithm itself is of far too great complexity to be of practical use, with a time complexity of \( O(L^{3N}) \) and a memory usage in the order of \( O(L^{2N}) \) given \( N \) sequences of length \( L \). The dynamic programming algorithm uses a \( N \)-dimensional matrix \( C \) to calculate the optimal cost of aligning and folding subsequences of the \( N \) sequences. One assumption in the approach is that compatible structural parts from all sequences are aligned, or that the substructure is completely deleted in a sequence. The cost function introduced by Sankoff is a weighted combination of the free energy of the structure and an alignment cost. To find the optimal alignment and structure, a number of auxiliary matrices are needed: \( D \) is used to keep track of the optimal alignment cost, \( F \) contains the free energy of the secondary structure, and \( G \) is needed when a multiloop is part of the structure. The calculations are complicated even for the simplest case of two RNA sequences, and for the general case of \( N > 2 \) sequences the approach quickly becomes infeasible.

Since the full Sankoff algorithm is intractable, different restricted versions have been developed either limiting the number of sequences, simplifying the cost scheme, or abandoning the search for an optimal solution.

One of of first attempts at implementing a restricted Sankoff algorithm is FOLDALIGN [GHS97, HLSG05] which is limited to pairwise alignment. The program performs local
alignment based on the Smith-Waterman algorithm, and while FOLDALIGN was initially limited to Nussinov-style base pair maximization, a more complex energy model was added later. The scoring scheme is dynamic and considers the structural context – i.e. is it a stem, a hairpin loop, a bulge etc. The score includes both energy, stacking and substitutions. Some further simplifications are introduced to reduce the running time.

The length of the local motifs being aligned is limited to be at most $\lambda$ nucleotides, and the two motifs cannot be more than $\delta$ nucleotides apart. With these restrictions, the time complexity becomes $O(L^2\delta^2\lambda^2)$.

The original FOLDALIGN algorithm did not consider branching and can be seen as a more complex version of the Nussinov recursion given in Eq. 3. A $25 \times 25$ substitution matrix is used, containing the score of aligning all possible pairs of letters $a \in \Sigma_G$ considering both sequence alignment and base pairing potential.

The dynamic programming is done in a 4-dimensional matrix $D$, where entry $D_{i,j;k,l}$ contains the optimal score of aligning the subsequence from nucleotide $i$ to nucleotide $j$ in one sequence with the subsequence from nucleotide $k$ to nucleotide $l$ in the other. Originally, the recursion was done over 15 possible cases for each entry, taking into account whether the ends of the two subsequences were aligned, whether gaps were inserted etc. To include branching structures, a recursion similar to the Nussinov branch given in Eq. 4 was included in [HLSG05].

The Dynalign program by Mathews and Turner [MT02,Mat05] is another algorithm based on the Sankoff approach. Similar to FOLDALIGN, Dynalign performs pairwise alignment although it finds a global solution. The scoring scheme is based on energy minimization but does not consider sequence alignment, meaning that the predicted structure is optimized while the alignment of unpaired loops and bulges is arbitrary.

A different approach is taken in the pairwise program PMcomp and its multiple sequence variant PMmulti [HBS04]. For two sequences $A$ and $B$, PMcomp aligns the base pair probability matrices, $P^A$ and $P^B$, calculated for each sequence using the partition function described previously [McC90]. PMcomp takes both sequence alignment and structure conservation into account while finding the optimal structure $S^*$. The structure is defined as the set of aligned base pairs from the two sequences $(i,j;k,l)$. The unpaired nucleotides are not part of the structure but are instead aligned using a score function $\sigma$:

$$ S^* = \arg\max_{S \in S} \left( \sum_{(i,j;k,l) \in S} (\Psi^A_{i,j} + \Psi^B_{k,l} + \tau(a_i, a_j; b_k, b_l)) + \gamma N_G + \sum_{i,k \notin S} \sigma(a_i, b_k) \right) $$

The sum over all aligned base pairs $(i,j;k,l)$ in the structure combines the base pair probabilities with a score for aligning the nucleotides given by $\tau$. The probability matrices are log-transformed into scores $\Psi$ using a probability threshold $p_{min}$: $\Psi_{i,j} = \log \left( \frac{P_{i,j}}{p_{min}} \right)$. Gaps are included using the linear gap penalty $\gamma$, and the last sum scores the aligned, unpairing nucleotides.
The search through the solution space $S$ is again carried out using dynamic programming, using a matrix $S$ to keep track of the optimal score of aligning subsequences from $A$ and $B$, and a similar matrix $S^M$ with the constraint that the terminal nucleotides are aligned base pairs. The recursions used are similar to the ones seen previously.

Multiple alignment and structure prediction is carried out by PMmulti using a progressive approach: All-against-all pairwise alignment using PMcomp is performed, and the closest pair of matrices, $P^A$ and $P^B$, is located. These two matrices are combined into a new $L \times L$ matrix $P^{AB}$, where $L$ is the length of the alignment. The new matrix defines the likelihood of having a base pair between any pair of columns in the alignment. If either column contains a gap, a base pair cannot form and the entry is 0. Otherwise, the base pair probability is set to the geometric mean of the probabilities of the involved base pairs.

Other more recent programs follow the approach of PMmulti. In FOLDALIGNM [THG07], various improvements to time and memory usage are introduced: They introduce the FOLDALIGN parameter $\delta$ that limits the distance between aligned subsequences, thereby improving the running time to $O(L^2\delta^2)$. Additional optimizations described in [HTG07] are introduced, limiting the number of entries in the dynamic programming matrix that need to be calculated. Also, the program makes it possible to use base pair probability matrices calculated using FOLDALIGN instead of the partition function.

The multiple alignment is carried out similarly to PMmulti, but the aligned probability matrices are combined differently. The PMmulti approach excludes any base pair that is not completely conserved by setting the entry to 0 as seen above. In FOLDALIGNM, these entries are instead set to the fraction of sequences that can form a base pair at that position if there is enough evidence (more than 40% of the sequences in the alignment). The final structure is therefore not a strict consensus since some sequences might miss some of the base pairs, but for large alignments the approach is shown to work.

LocARNA [WRH+07] was implemented to be able to deal with genome-scale datasets, i.e. it has to be able to work with thousands of sequences and should be robust against sequences being included in the set by mistake. The algorithm mimics PMcomp but with some improvements: First of all, LocARNA performs local alignment which is useful in genomic scans. It also limits the number of significant entries in the dynamic programming matrix, thus speeding up computations. In a manner similar to FOLDALIGNM, the combination of probability matrices is also altered to avoid setting most entries to 0.

An entirely different approach to the problem of simultaneous multiple alignment and structure prediction that deserves mention is taken in LARA [BKR07]. Here, alignment and structure prediction is cast as a graph theoretical problem which is then solved as an integer linear program. The graph representation of a structural alignment contains vertices $V$ for the nucleotides and edges that represent alignment of nucleotides, interaction edges (i.e. base pairs) and gaps in the alignment. A maximization problem is formulated under constraints from a set of inequalities that enforce that e.g. base pairs
do not cross. Since the problem is \( \mathcal{NP} \)-hard, Lagrangian relaxation is used to make the solution feasible.

The program CMfinder [YWR06] uses covariance models (CM) [DEKM98] to perform the alignment and structure prediction. The procedure is aimed at finding RNA motifs and thus performs local alignment. In CMfinder, probabilistic models for sequence motifs and RNA structure are combined, inspired by the DNA motif search program MEME [BE95] and the CM for RNA structure COVE [ED94]. First, stable substructures for all sequences are located using the Vienna package [HFBS94], a set of structures conserved between all sequences is found, and this set of subsequences is aligned heuristically. This initial alignment and structure is then iteratively refined using expectation-maximization. One advantage with CMfinder is that the probabilistic model generated in the process can be directly used for homology search, thus allowing for iterative refinement of the model by including new members into the alignment.

In the paper “MASTR: multiple alignment and structure prediction of non-coding RNAs using simulated annealing” presented in chapter 2, we outline the method we have developed for simultaneous multiple alignment of RNA sequences and prediction of a common secondary structure. In MASTR, we take an approach using simulated annealing to iteratively improve on an initially random alignment, slowly finding a stable multiple alignment. Interspersed with the changes to the sequence alignment, changes to the predicted secondary structure are performed leading to the simultaneous optimization of both alignment and structure. The method is shown to perform comparatively well to other methods. In the following the method will be described in some more detail.

The process in MASTR is based on sampling from the solution space, starting with some initial – possibly random – alignment. A Markov chain Monte Carlo (MCMC) approach is taken [And03, Häg02], where we link the cost of the structural multiple alignment to a probability distribution. Since the optimum of this probability distribution cannot be found analytically – otherwise making the sampling unnecessary – the solution space is sampled while trying to avoid being caught in local minima. The sampling is done by changing the alignment and/or structure, thus moving between possible solutions based on the cost function used.

Unless otherwise specified, the sequences are assumed to be unaligned, and a random alignment is generated by inserting gaps throughout the sequences. Since the initial alignment is random, the sampling steps in the early stages can be comparatively large in order to perform a proper search. Especially early on, it should also be possible to take steps that momentarily increase the cost to escape from local minima. As the sampling continues, and we converge on a solution, only steps improving the overall cost are accepted. We carry out the sampling using simulated annealing [KGV83, LEB92].

The sampling procedure of simulated annealing is inspired by physics and the formation of crystals by cooling. The atomic positions in that case can be modeled as a probability distribution dependent on the temperature of the system. By lowering the
temperature slowly, a structure close to the minimum free energy of the crystal is found, and by similarly lowering an artificial temperature in the sampling process, the hope is to find a solution close to the optimum. It can be shown that for a finite problem, simulated annealing will approach the optimal solution given infinite time [Håg02].

In the present case of structural alignment of RNAs, the simulated annealing works by making small changes to the alignment and/or structure. When a change is made, the cost of the alignment might change. If the cost after the change is the same or lower, the new alignment is kept since it is at least as good as the previous state. If the cost is increased, the change might be rejected. As stated above, making seemingly bad moves might be necessary to escape from local minima. Whether the new state is accepted or not depends on the iteration and the increase in cost – if the increase is large, the probability of acceptance should be correspondingly low. Also, after many iterations when we approach the final solution, changes that increase the cost should be rare.

The simulated annealing uses an artificial temperature $T$ to govern the sampling. The initial $T$ should be sufficiently high to allow many changes, and over time the temperature decreases as the alignment becomes more and more stable. The probability of accepting a change to the structural alignment is calculated using the Metropolis–Hastings algorithm [Has70, KGV83], which depends on the current temperature $T$, the cost of the previous state $c_{\text{OLD}}$ and the cost of the new state $c_{\text{NEW}}$:

$$P = e^{-\frac{(c_{\text{NEW}} - c_{\text{OLD}})}{T}}$$  \hspace{1cm} (5)

Thus, MASTR samples the solution space based on the cost function, and since a state only depends on the previous state – given the set of possible moves – this generates a Markov chain. In MASTR, we use an exponentially decreasing temperature [LEB92], where the temperature $T_i$ at the $i$'th iteration is based on the temperature at the previous step $T_{i-1}$ and a scaling factor $\tau$:

$$T_i = T_{i-1} \cdot \tau \quad 0 < \sigma_{\text{exp}} < 1$$

The number of iterations $I$ can be given as an argument to MASTR, or it can be set internally based on the size of the alignment given as the number of nucleotides, $N_{\text{TOTAL}}$, and the length of the longest sequence, $L_{\text{MAX}}$:

$$I = \alpha N_{\text{TOTAL}} + \beta L_{\text{MAX}}$$

The factors are by default set to $\alpha = 1000$ and $\beta = 1700$. We set the initial temperature and scaling factor based on the number of iterations and the actual dataset. In the first 0.1% of the iterations all changes are accepted and the results are used to estimate the standard deviation $\sigma$ of the cost function. By defining an initial probability of acceptance $P_0$ (0.99 by default), the initial temperature is set to be:

$$T_0 = -\frac{\sigma}{\log_2(P_0)}$$
The final temperature should be close to 0 for the system to approach the optimum, giving the scaling factor needed:

\[ \tau = \exp\left( \frac{\log_2 \left( \frac{10^{-5}}{T_0} \right)}{I} \right) \]

where we set the final temperature to be 0.00001.

The sampling procedure searches through the solution space by making local changes to the alignment and/or structure. These moves are divided into those aimed at changing the structure, and those aimed at changing the alignment (perhaps also affecting the structure).

The sequence moves to alter the alignment can only be done through manipulating the gaps. In MASTR, we have three simple moves to this effect: Move blocks of gaps, adding columns of gaps, and removing columns of gaps. A gap block is defined to be a local part of the alignment containing nothing but gaps, spanning \( H \) positions and including \( V \) sequences. A gap block is found by randomly choosing a gap in a random sequence and expanding the \( 1 \times 1 \) gap block with some probability in both directions, if possible. The gap block is then moved to a new position in the alignment, and the cost is updated. Columns of gaps can also be added to either end of the alignment, from where gaps can be moved to other positions, just as columns of gaps can be removed entirely from the alignment to condense it.

The structure moves are equally simple: A new base pair between columns \( i \) and \( j \) is proposed, and it is added to the structure if neither position is involved in another base pair, and if the new base pair does not cross an existing one. Base pairs can also be added by choosing a random base pair already in the structure and adding a new base pair on either side of the chosen base pair or the stem of which it is a part. The structure can be reduced by randomly removing a base pair from the structure.

The cost of the structural alignment includes a structure part and a sequence alignment part. The cost of the sequence alignment uses the entropy given in Eq. 2 using a word size of 1, i.e. single nucleotides. The probability of the multiple alignment \( MA \) can be conveniently illustrated as a Hidden Markov Model as shown in Fig. 4. The transitions in the model are given by the type of letter in the previous state: The probability is \( P_{GO} \) and \( P_{GE} \) for opening and extending a gap, respectively, (moving through the circle-states in Fig. 4), while the transition probabilities are \((1 - P_{GO})\) and \((1 - P_{GE})\) when nucleotides are emitted (moving through the square states).

In the gap-states, only one character is possible giving an emission probability of 1, while the emission probabilities in the nucleotide states depend on the actual nucleotide. The probability of seeing the nucleotide \( x \) at position \( j \) in sequence \( i \), \( P(x_j^i) \), is the observed frequency of that particular nucleotide in the alignment column:

\[ P(x_j^i) = \frac{c_j(x_j^i)}{\sum_{x' \in \Sigma} (c_j(x') + \pi(x'))} \]
where the function $c_j(x)$ counts the number of occurrences of nucleotide $x$ at position $j$ in the alignment, and $\pi(x)$ is a pseudocount function. In MASTR, we use the simple $\pi(x) = 1$ for all $x$.

Given the probability of the alignment, the entropy is calculated as the negative logarithm of $P(\text{MA})$. The cost function, $\text{cost}_H(\text{MA})$ is defined as the entropy of the alignment, $H(\text{MA})$, normalized by the total number of nucleotides in the alignment, $N_{\text{TOTAL}}$:

$$\text{cost}_H(\text{MA}) = \frac{H(\text{MA})}{N_{\text{TOTAL}}}$$

The cost of the structure contains two quantities: The energetic stability is measured using the partition function [McC90], and the evolutionary evidence is measured using a modified version of the covariation used in RNAalifold [HFS02].

The RNAfold program from the Vienna package [HFBS94] is used to calculate the base pairing probabilities for each sequence. When a base pair is proposed between two alignment columns, the probability $P^{\text{MA}}(i,j)$ that it is part of the consensus structure is the average probability of seeing the base pair in the individual sequences. If a sequence contains a gap in either position, the base pair probability for that sequence is 0. The cost of the base pair is then set based on a background probability $P_{\text{null}} = 0.25$:

$$\text{cost}_P(i,j) = -\log_2 \left( P^{\text{MA}}(i,j) \right) + \log_2 \left( P_{\text{null}} \right)$$

The covariation measure $B_{i,j}$ used in RNAalifold is modified to include stacking of base pairs, which was shown to discriminate well between true and false base pairs in the paper presented in chapter 1. This is done by including the two surrounding column pairs in the calculations while still putting most emphasis on pair $(i,j)$:

$$C(i,j) = \frac{B_{i-1,j+1} + 2 \cdot B_{i,j} + B_{i+1,j-1}}{4}$$

**Figure 4**: The probabilistic model of the multiple alignment illustrated as a Hidden Markov Model. The alignment starts in the diamond shaped state, the circule states emit gaps, while square states emit a nucleotide $N \in \Sigma$. 
As with the base pairing probabilities, this measure is turned into a cost function by introducing a threshold parameter $\Phi = 0.25$:

$$\text{cost}_C(i, j) = -C(i, j) + \Phi$$

The full cost of the structural alignment is a weighted combination of these three terms – the entropy cost function, the base pair probability cost, and the covariation cost – using the weights $\alpha$ and $\beta$, thus combining the evidence for the alignment $MA$ and the structure $S$ in a way that makes it possible to sample both using the described simulated annealing approach:

$$\text{cost}(MA, S) = \text{cost}_H(MA) + \sum_{(i, j) \in S} (\alpha \text{cost}_P(i, j) + \beta \text{cost}_C(i, j))$$  \hspace{1cm} (6)

The default weights in MASTR are $\alpha = 1.5$ and $\beta = 0.6$. In the paper presented in chapter 2, the method is tested against other programs and shown to perform comparatively well.

**Combining approaches**

In the previous sections, we have seen various approaches to solving the RNA structure prediction problem – including our own MASTR-program. It is evident from the various methods that researchers have approached the problem from different angles, giving each method its own strengths and weaknesses. Also, a more mundane observation is the fact that while it can be difficult to enough to run your own programs, it can be a nightmare to get other peoples programs to work – especially if you are a biologist with limited computer knowledge or just working on a Windows machine.

Since experience tells us that the latter is in fact often the case, a different approach is needed if we as bioinformaticians in general want our programs to be used by actual researchers in the labs. They have to be easy to use, and the results should be easy to understand. Also, if different comparable methods exist, it would be useful to easily compare predictions to check for errors or to confirm the prediction if there is an overlap. Such an approach poses several challenges: It should require only limited computer knowledge, it should be platform independent, and it should be easy to submit your data and see the results.

We decided to approach this task for the case of RNA alignment and structure prediction by installing the top programs in the field on a computer cluster. We set up a framework that automatically transformed the input data from standard Fasta format to the required format for the individual programs, and called each program with the submitted data. The output from all programs is then parsed and analyzed in various ways before being presented to the user in a standardized manner along with downloadable files in different widely used formats.
This approach is presented in the paper “WAR: Webserver for aligning structural RNAs”. Here, we have made a webserver that makes it possible to easily run various methods on your set of RNA sequences\(^1\). The sequences are analyzed by all the chosen methods and a consensus prediction is also presented. Each alignment and structure is evaluated using different measures such as average base pairing free energy, covariation etc. The webserver has been widely used since it was made available. A local version of WAR that I implemented during my stay at the Wellcome trust Sanger Institute has been used in the latest version of Rfam [GDT\(^+\)08], and the WAR webserver has also been used in the analysis of new ncRNAs of unknown structure [KTR\(^+\)09].

\(^1\)As of this writing, the webserver is hosted at the Faculty of Life Sciences and is unavailable due to some mysql error. During the spring of 2010, it will be moved to a different location where it can be properly maintained.
Bibliography


Chapter 1

Measuring covariation in RNA alignments: Physical realism improves information measures

Stinus Lindgreen, Paul P. Gardner and Anders Krogh

Measuring covariation in RNA alignments: physical realism improves information measures

S. Lindgreen*, P. P. Gardner and A. Krogh

Bioinformatics Centre and Molecular Evolution Group, Institute of Molecular Biology, University of Copenhagen, Universitetsparken 15, 2100 Copenhagen Ø, Denmark.

Received on June 28, 2006; revised and accepted on October 4, 2006
Advance Access publication October 12, 2006
Associate Editor: Golan Yona

ABSTRACT

Motivation: The importance of non-coding RNAs is becoming increasingly evident. Mutations in the DNA and protein levels are not the only measure of change. RNA structures are becoming increasingly important. Knowledge of the secondary structure is useful for RNA folding and the addition of conserved motifs.

Methods: We use two methods to calculate covariation in RNA alignments. The first is RNAalifold, which integrates the Sankoff algorithm with a detailed energy model. The second is RNAalifold covariation, which uses a modified energy model that includes stacking.

Results: We compare these methods to the RNAalifold program, which uses a detailed energy model. We find that the RNAalifold covariation method produces more accurate predictions than the RNAalifold program.

Availability: Scripts, data and supplementary material can be found at http://www.binf.ku.dk/Stinus_c covariation

Supplementary information: Supplementary data are available at Bioinformatics online.

1 INTRODUCTION

In recent years, the role of non-coding RNAs has become increasingly evident. Many RNA structures play a more active role in the cell than just being the mediator of information between the DNA and protein levels. These non-coding RNAs (ncRNAs) have a functional role, and their function is frequently tied to the structure of the molecule. The comparative approach relies on two conflicting properties: information and alignment quality. Structure can only be inferred for positions that have actually mutated. Fully conserved columns have no covariance information, and thus highly diverged sequences carry most covariance information. On the other hand, highly diverged sequences are difficult to align correctly, and therefore the comparative methods are expected to be best for sequences that are diverged to the point where one can still obtain a good multiple alignment (without using covariance information). A recent survey showed that sequences below ~65% identity were inaccurately
aligned, thus destroying secondary structure information (Gardner et al., 2005). It is also clear that it may be advantageous to use covariance information for the alignment, and thus align and predict structure at the same time, but current methods for that are limited to small alignments.

As more focus is being given to ncRNAs, interest in locating these in genomic sequences is growing. Genefinders for this problem are appearing: Using stochastic context-free grammars is employed in the programs QRNA (Rivas and Eddy, 2001) and EvoFold (Pedersen et al., 2006). In these, the grammars are designed to detect signal from base pairing interactions using an implicit covariation measure. MSARI (Coventry et al., 2004) uses a combination of base pairing probabilities (McCaskill, 1990) and a sliding window for finding complementary subsequences while allowing for small misalignments. In RNAz (Washietl et al., 2005) a sliding window is used to fold subalignments using RNAalifold. The consensus structure is compared to the minimum free energy structures of the individual sequences. The covariation measure used in this approach is therefore the same as in RNAalifold.

To measure covariation between two sites in the RNA molecule, where the sequence is changed while base pairing interactions are preserved, mutual information is the textbook example, e.g. (Durbin et al., 1998) because it measures the information in base pairs which cannot be explained from single base frequencies. The idea is to find sites where the degree of co-occurring mutations is higher than one would expect by chance. Sites showing a high degree of covariation are seen as likely base pairs. Several methods include an MI component in their scoring scheme, e.g. COVE (Eddy and Durbin, 1994), ILM (Ruan et al., 2004), MatrixPlot (Gorodkin et al., 1999), KNetFold (Bindewald and Shapiro, 2006) and ConsStruct (Lück et al., 1999).

Although MI and other measures of covariation are widely used there has, to the best of our knowledge, never been a thorough analysis of the discriminative power of these measures. In this study, we analyze a number of RNA datasets using different measures of covariation. The datasets cover both different classes of structural RNAs and different degrees of overall sequence identity. We show that the standard MI is not very discriminative, and that extending the measure with additional structural information yields a more powerful measure. The best covariation measure tested in this study is a new formulation of the measure used in RNAalifold (Hofacker et al., 2002) where stacking of base pairs is taken into account. The most discriminative measure, though, is averaged base pairing probability matrices calculated using the partition function (McCaskill, 1990), which uses energy terms and is independent of covariation information.

### 2 APPROACH

We have evaluated the discriminative power of different covariation measures. The following measures described previously in the literature were implemented:

- Standard MI (Shannon, 1948; Chiu and Kolodziejczak, 1991; Gutell et al., 1992)
- MI summing only Watson–Crick and wobble base pairs (Gorodkin et al., 1999) (MI$^{w}$)
- Normalized MI (Martin et al., 2005) (MI$^{N}$)
- The covariation measure used in RNAalifold (Hofacker et al., 2002) (MI$^{w,C5}$)
- MI summing base pairs and including stacking (MI$^{w,C5,5}$)
- MI summing base pairs, including stacking and using gap penalties (MI$^{w,C5,5,p}$)
- MI summing base pairs and using gap penalties (MI$^{w,C5,5,p,W}$)
- The B measure including stacking ($B^{w}$)

A number of novel measures based on the preceding list were implemented and evaluated as well:

- MI using gap penalties (MI$^{p}$)
- MI summing base pairs and including stacking (MI$^{w,C5,5,p}$)
- MI summing base pairs, including stacking and using gap penalties (MI$^{w,C5,5,p,W}$)
- MI summing base pairs and using gap penalties (MI$^{w,C5,5,p,W}$)

Note that for simplicity some of the measures are denoted MI although they are technically not mutual information measures. This will be elaborated in the following. The performance of the above measures is compared to using averaged base pairing probability matrices calculated using the partition function (McCaskill, 1990). We used the implementation from the Vienna package (Hofacker et al., 1994). The different measures are described in detail under methods.

The datasets were compiled by making three random samplings from each of three large structural alignments (tRNA, 5S rRNA and U5) (Griffiths-Jones et al., 2003; Szymanski et al., 2002; Zwieb, 1997) yielding a total of nine datasets. The alignments used in this study are known to be of high quality, and they have previously been used in the benchmark by Gardner et al., 2005, but other datasets could in principle have been used. The sampling was performed in such a way that the overall identity of the subalignments was controlled. The subalignments are of low (40–60%), medium (60–80%) and high (80–100%) overall identity, where the % ID is calculated using the reference alignments. In each subalignment, any all-gap columns were removed but otherwise the correct alignment and reference structure is preserved. The sizes of the alignments are summarized in Table 1.

The nine individual datasets were analyzed separately, but in the following the results reported are averages for each of the identity intervals (i.e. containing an alignment of 5S rRNA, tRNA and U5). This is done to avoid bias from the composition of the individual families while measuring the performance as a function of sequence similarity.

The measures are compared using the Matthew’s correlation coefficient (MCC, see methods), which is maximal ($\geq 1$) if, for a given threshold $y$, all true base paired columns are above and all other pairs are below $y$. Thus, the higher the MCC, the better the discrimination. Each measure was evaluated using 100 threshold values evenly distributed between the minimum and maximum value for that particular measure. Since the range and distribution...
of the different measures vary, the actual threshold values are not comparable. Other binning strategies were tested but did not affect the results (data not shown).

3 METHODS

The covariation measures used in this study are described in the following along with the evaluation scheme used to benchmark the measures.

3.1 Evaluating the measures

Each covariation measure is evaluated based on how well it discriminates between true and false base pairs. Given a sequence alignment of structural RNAs, every possible base pair \((i, j)\) can be evaluated using each of the suggested measures. Since the reference structure is available, these scores can be divided into true base pairs \(BP_T\) and false base pairs \(BP_F\).

Each measure calculates a score for a possible base pair, but the question is how large a score has to be to best discriminate true base pairing interactions from false. For this, a threshold value has to be used. For a given threshold value, \(\gamma\), the number of true positives, \(TP\), can be found as the number of scores in \(BP_T\) greater than or equal to \(\gamma\). Similarly, the number of false negatives, \(FN\), is the number of scores smaller than \(\gamma\). The number of false positives, \(FP\) and true negatives, \(TN\), can be found in a similar manner from the scores in \(BP_F\).

To evaluate the different measures, 100 thresholds evenly distributed between the minimum and maximum score were used and the numbers \(TP\), \(FP\), \(TN\) and \(FN\) calculated. For example, for the standard mutual information, 100 numbers between 0 and \(\text{log}_2(4)\) were tried. For each threshold, one could use the sensitivity and positive predictive value to evaluate the discriminative power of the measure. Instead of selecting thresholds based on either of these, however, a balance between them is sought by using the MCC (Matthews, 1975):

\[
MCC = \frac{TP \times TN - FP \times FN}{\sqrt{TP + FP}(TP + FN)(TN + FP)(TN + FN)}
\]

Using this, MCC as a function of the threshold values was analyzed for each covariation measure (see Supplementary material). The actual values of the thresholds vary between the measures due to the different ranges, but an equal number of possible thresholds was used for each.

The datasets analyzed contained either sRNA, SS sRNA or US sequences. For each of these families, three datasets were generated having an overall identity of 40–60%, 60–80% and 80–100% (in the following these are referred to as low, medium and high identity, respectively). For each identity interval the average over the three datasets was used, thus showing the dependency on the sequence similarity while being independent of the specific family.

3.2 Standard MI

Functional RNA molecules are under selective pressure to preserve their secondary structure. This evolutionary pressure can lead to a change in the primary sequence that keeps the base pairing interactions intact. An example of such a chain of mutations is:

\[
A \Rightarrow U \rightarrow G \rightarrow U \rightarrow G \rightarrow C
\]

Thus, related RNAs can differ in sequence while having the same structure. This makes pure sequence alignment of structural RNAs difficult. As shown in Gardner et al. (2005), most sequence alignment tools fail when the sequence identity is below 50–60%. To use this evolutionary information in the prediction of secondary structure, one needs to find pairs of columns in the alignment that show a higher degree of covariation than expected by chance. This might indicate base pairing interaction between the two positions. The classic measure for this is the MI content (Shannon, 1948; Chiu and Kolodziejczak, 1991; Guttell et al., 1992; Durbin et al., 1998).

In the context of RNA alignment, this score is defined on pairs of columns, \(i\) and \(j\), in the multiple alignment. Let \(a\) be a letter from column \(i\) and let \(b\) be a letter from column \(j\). The frequency \(f_i(ab)\), at which a base pair of type \(ab\) is observed is compared to the number of times one would expect the pair to occur by chance. The latter is calculated using the frequencies of the two single nucleotides in the two columns, \(f_i(a)\) and \(f_j(b)\). For two columns \(i\) and \(j\), the mutual information is given by the following expression:

\[
MI_{ij} = \sum_{ab} f_i(ab) \log \frac{f_i(ab)}{f_i(a)f_j(b)},
\]

where the sum is over all 16 possible pairs of bases in the two columns. Keep in mind that \(0 \log(0) = 0\). This is the relative entropy of the joint distribution relative to the product distribution, also known as the Kullback–Leibler divergence (Kullback and Leibler, 1951). The MI gives the amount of information obtained about one position in the alignment if one knows what the other position is. The higher the MI, the more information is gained.

There are some problems with this measure: if one or both positions are conserved in sequence the MI is zero. Another problem is the level of noise from covarying columns that cannot form base pairs and thus should not contribute to the measure in the context of classical canonical pairing. Furthermore, structurally neutral mutations, such as \(A \rightarrow U \rightarrow G \rightarrow U\) do not contribute to the MI. Finally, there is the question of how to deal with gaps in the alignment.

Since gap characters symbolize insertion/deletion events, it makes little sense to deduce covariation from them. It is therefore necessary to subtract the number of gaps when calculating the MI. Note, however, that the number of gaps in the two columns \(i\) and \(j\) may vary. But if the frequencies are based on two different numbers of observations, the MI score fails. Instead, all the pairs that contain at least one gap character are disregarded when calculating frequencies.

3.3 MI with gap penalty

As mentioned, the standard formulation of MI does not penalize gaps. Instead, positions with gaps are disregarded. This presents a new problem: consider a pair of columns containing many gaps but where the few remaining positions display a high degree of covariation. Since the gaps are disregarded, this column pair receives a good score. This is not necessarily the desired behaviour: it would be interpreted as a high degree of structural conservation, but that does not correspond with the large number of indels.

Instead, a large number of gaps at a given position implies that the region is variable and the MI score is less certain. This means that less weight should be placed on these positions. A simple way to incorporate gap penalties into the MI score is to define a gap penalty \(\beta\) and let it influence the MI score as a function of the number of gaps positions. Let \(N_G\) be the number of positions containing at least one gap, and let \(MI_{ij}\) be defined as in Equation (1). A variation of the MI score with gap penalties is:

\[
MI'_{ij} = MI_{ij} - N_G \cdot \frac{\beta}{2}
\]

As can be seen, the MI score of a column pair with no gaps will not be penalized, while more gaps give a larger penalty. If all positions contain a gap, the column pair will receive a negative MI score of \(-N_G \cdot \beta\). In the experiments we used \(\beta = \frac{1}{4}\).

3.4 MI using only canonical base pairs

The standard MI is sensitive to noise from unwanted, non-canonical base pairs. A possible variation of MI that might limit the noise is to focus on the acceptable base pairs alone and ignore the rest (Gorodkin et al., 1999). While the 4 nucleotides give rise to 16 possible base pairs, only 6 of these are considered structurally important. If this distinction is incorporated into the MI score, it would only gather information from positions that actually display structural covariation.

Let the set of the six canonical base pairs be denoted \(BP\). Using Equation (1) over members of the set \(BP\) instead of all 16 possible pairs would not
yield a mutual information since only a subset of the possible occurrences is used. Instead, a relative entropy measure (Cover and Thomas, 1991) is introduced, but for consistency the MI terminology is used.

Let $b_{ij} \in \{0, 1\}$ indicate whether there is a canonical base pair between columns $i$ and $j$, and let $p(b_{ij} = 1)$ be the probability of such a base pair. Then, the probability of having a canonical base pair between columns $i$ and $j$ becomes:

$$p(b_{ij} = 1) = \sum_{ab \in B_p} p_a(p(b_{ij} = ab)).$$

The probabilities of the specific base pairs can be estimated from the actual frequencies. This probability is compared to the probability of observing a base pair by chance given the single nucleotide probabilities. Let this background be denoted $q(b_{ij} = 1)$:

$$q(b_{ij} = 1) = \sum_{ab \in B_p} p(a)p(b).$$

This incorporates only the six canonical base pairs. Using these definitions, the probability for not having a base pair is given as $p(b_{ij} = 0)$ with background $q(b_{ij} = 0)$. In combination, this yields the following relative entropy:

$$\text{MI}_j^p = \sum_{x \in \{0, 1\}} p(b_{ij} = x) \log \frac{p(b_{ij} = x)}{q(b_{ij} = x)},$$

Note that $p(b_{ij} = 0)$ and $q(b_{ij} = 0)$ are equal to $1 - p(b_{ij} = 1)$ and $1 - q(b_{ij} = 1)$, respectively. Although this is the most rigorous measure, experiments showed that better discrimination was achieved if the term with $x = 0$ was dropped from the measure (data not shown). The final scoring function therefore only considers the structural base pairs, thus filtering noise from non-pairing interactions:

$$\text{MI}_j^p = p(b_{ij} = 1) \log \frac{p(b_{ij} = 1)}{q(b_{ij} = 1)}$$

Intuitively, this measure avoids some of the problems with the standard MI by explicitly focusing on the canonical base pairs. A measure using only the log-odds score ($\log \frac{p(b_{ij} = 1)}{p(b_{ij} = 0)}$) was also tried but it did not perform well (data not shown).

A combined measure, that includes an explicit gap penalty, can easily be defined in a manner similar to Equation (2):

$$\text{MI}_{M^p}^p = \text{MI}_j^p - N_{ij}^p \beta$$

### 3.5 MI with stacking

The stacking of adjacent base pairs—also known as nearest-neighbour interactions—is a common feature in RNA secondary structure (Borer et al., 1974; Otsu and Tinoco, 2004). Therefore it is reasonable to extend the MI measure and incorporate stacking. Using column pair $(i, j)$ as a reference, if MI, is large, stacking implies that the adjacent column pair $(i + 1, j - 1)$ might also give a good score. By combining the two expressions, stacking would be considered explicitly by the MI.

As mentioned for the standard MI, there is a problem with noise, and this is dramatically increased when considering adjacent columns. In the standard formulation of MI, 16 pairs are considered of which 10 are non-canonical. In the stacking formulation, the combination of two columns lead to a summation over 256 terms of which 220 contain at least one non-canonical pair. Thus, the signal-to-noise ratio decreases.

Instead, a measure incorporating stacking but only considering the canonical base pairs is used. This drastically reduces the number of variables to estimate, which makes it possible to calculate the measure based on most alignments. The measure uses relative entropy and is an extension of the MI$^p$ measure. Let $b_{ij} \in \{0, 1\}$ be defined as before, and let $b_{i+1,j-1} \in \{0, 1\}$ indicate a base pair at the internal positions. The relative entropy is a sum over the four possible combinations of $b_{ij}$ and $b_{i+1,j-1}$ (corresponding to pair/pair, pair/not-pair, not-pair/pair and not-pair/not-pair):

$$\text{MI}_{M^p}^p = \sum_{x \in \{0, 1\}} p(b_{ij} = x, b_{i+1,j-1} = y) \log \frac{p(b_{ij} = x | b_{i+1,j-1} = y)}{q(b_{ij} = x | b_{i+1,j-1} = y)}$$

The probabilities are estimated from the aligned sequences. For a given column pair in an alignment, let $c_{ij}$ count the number of times that $b_{ij} = x$ and $b_{i+1,j-1} = y$. For instance, $c_{11}$ is the number of times a canonical pair is observed both between $(i, j)$ and $(i + 1, j - 1)$. To calculate the relative entropy, only these four numbers are necessary making it a practical measure to use. The joint probability is simply found using the corresponding count:

$$p(b_{ij} = x, b_{i+1,j-1} = y) = \frac{c_{xy}}{N},$$

where $N$ is the number of sequences. The conditional probability is found using the relation:

$$p(a | b) = \frac{p(a, b)}{p(b)}$$

in the present case gives:

$$p(b_{ij} = x | b_{i+1,j-1} = y) = \frac{c_{xy}}{c_{y}} = \frac{c_{xy}}{c_y + c_{2x}},$$

where $c_y$ describes the probability of the observations occurring by chance. The dinucleotide probabilities are estimated from the single nucleotide frequencies. Since $q$ is the null-model, the columns are considered independent. Therefore, the conditional probability $q(a | b)$ is simply the probability of the first random variable:

$$q(b_{ij} = x | b_{i+1,j-1} = y) = q(b_{ij} = x)$$

Practical experiments showed that the measure performed best when only considering positions containing a canonical base pair between columns $i$ and $j$. This corresponds to always demanding $b_{ij} = 1$, effectively removing the outer summation. This corresponds well to the final formulation of MI$^p$. The stacking measure using canonical base pairs is:

$$\text{MI}_{M^p}^{p,\text{stack}} = \sum_{x \in \{0, 1\}} p(b_{ij} = 1, b_{i+1,j-1} = y) \log \frac{p(b_{ij} = 1 | b_{i+1,j-1} = y)}{q(b_{ij} = 1)}.$$ 

This measure can be extended with an explicit gap penalty as in Equation (2):

$$\text{MI}_{M^p}^{p,\text{stack}} = \text{MI}_{M^p}^{p,\text{stack}} - (N_{ij}^p + N_{i+1,j-1}^p) \beta.$$

Since the number of gap positions is found from two column pairs, the gap penalty $\beta$ is modified to fit the possibly larger number of gaps by using $\beta' = \beta/2$.

### 3.6 Normalized MI

Martin et al. (2005) argue that normalizing the standard MI score by the joint entropy of the same random variables yields a more discriminative measure. Given a multiple alignment, let $a$ be a character from column $i$ and let $b$ be a character from column $j$. The joint entropy of the two columns is given as:

$$H_{ij} = - \sum_{ab} P_{ij}(a, b) \log P_{ij}(a, b),$$

where $P_{ij}(a, b)$ is the joint probability of observing character $a$ in column $i$ and character $b$ in column $j$. The joint probability can be estimated from the
frequencies of the dinucleotides. This yields the following expression for the
normalized MI:
\[
\hat{M}_{i,j} = \frac{M_{i,j}}{\Pi_{i,j}}
\]
using the definition of \(\Pi_{i,j}\) from Equation (1).

It has been argued that this normalization removes some of the noise in the
MI score (Martin et al., 2005): a column pair with a large entropy score will also
receive a (not necessarily warranted) large MI score. This skew is
removed by normalizing with the entropy. However, this argument is based
on protein alignments and assumes that most of the pairs do not show
significant MI due to structural and functional constraints. This does not
necessarily hold for RNA alignments, where it is known that structure is
more conserved than sequence. Furthermore, the number of possible pairs is
much smaller in RNA than in proteins, so the MI signal might be better in the
present setting.

### 3.7 RNAalifold measure

In the RNAalifold program an alternative measure of covariation is used
(Hofacker et al., 2002). Let \(N\) be the number of aligned sequences, let \(a\) and
\(b\) denote sequences, \(a,b = 1,2, \ldots, N\), and let \(a_i^\alpha, b_i^\beta\) denote the character at
position \(i\) in sequence \(a\). As before, we only consider base pairs in \(BP\), i.e.
the set of Watson-Crick basepairs and the \(G + U\) wobble base pair. For each
sequence \(a\), the matrix \(\Pi^\alpha\) describes the possible base pairs. Thus, \(\Pi^\alpha_{i,j} = 1
\) if base pair \((a_i^\alpha, b_i^\beta)\) is in \(BP\), and \(\Pi^\alpha_{i,j} = 0\) otherwise.

Let \(C_{a,b,i,j}\) be the Hamming distance between two base pairs at
positions \(i\) and \(j\) in the alignment, i.e. \(\delta = 0\) if the 2 base pairs are identical,
\(\delta = 1\) if the base pairs vary at exactly one position (consistent substitution),
and \(\delta = 2\) if the 2 base pairs are different (compensatory mutations). The
more mutations observed that retain the base pairing interaction, the more
evidence that the base pair is correct:

\[
I(a_i^\alpha, b_i^\beta) = \begin{cases} 
0 & \text{if } \Pi^\alpha_{i,j} = 1 \\
1 & \text{otherwise.}
\end{cases}
\]

The penalty for the base pair under consideration is then found as:
\[
q_{i,j} = \frac{1}{N} \sum_{a,b} I(a_i^\alpha, b_i^\beta).
\]

The combined covariation measure is then given as:
\[
B_{i,j} = C_{i,j} - \delta q_{i,j}
\]
where \(\delta\) is a scaling factor for the penalty term. We used \(\delta = 1\) as in the
original paper.

### 3.8 The RNAalifold measure including stacking

The RNAalifold covariation already includes gap penalties and treatment of
canonical base pairs, and the idea of including stacking information can also
be extended to this measure. Originally, we used an asymmetric version that
only considered the base pair internal to \((i,j)\) cf. Equation (5). An anonymous
referee suggested the symmetric version described here which further
improved the performance.

For a pair of columns \((i,j)\), we also consider the neighbouring pairs
\((i−1,j−1)\) and \((i+1,j+1)\). Let \(C_{a,b,i,j}, \Pi_{i,j}^\alpha\) and \(I(a_i^\alpha, b_i^\beta)\) be defined as
above. The covariation for a pair \(a_i^\alpha, b_i^\beta\) in a sequence \(\alpha\) now also depends
on \(a_{i-1}^\alpha, b_{i-1}^\beta\) and \(a_{i+1}^\alpha, b_{i+1}^\beta\). The calculation of the covariation is, as before,
found by considering all possible sequence pairs, but now neighbouring
nucleotide pairs are considered. The inconsistency penalty \(q_{i,j}\) is found in a
similar manner.

By normalizing the stacking version to give a score in the same range as
the original measure, it becomes clear that the covariation measure \(B^s\) can be
found as a weighted average of the original RNAalifold score of the three
pairs under consideration. Thus, since the covariation between \((i,j)\) and
\((i+1,j+1)\) and between \((i,j)\) and \((i−1,j−1)\) is considered, the final formu-
lates becomes:
\[
B^s_{i,j} = \frac{B_{i-1,j-1} + 2 \cdot B_{i,j} + B_{i+1,j+1}}{4}.
\]

### 3.9 Base pair probabilities

The partition function described by McCaskill (1990) calculates the proba-
bility \(P(i,j)\) of seeing a base pair between positions \((i,j)\) in sequence \(s\) given
the nearest-neighbour energy model. The base pair probabilities are based
on the ensemble of all possible structures for the given sequence weighted by
the free energy of the individual structures. This information can be used to
assign probabilities to proposed base pairs in an alignment of \(N\) sequences as
follows: first, a probability matrix \(M^s\) is calculated for each unengaged
sequence \(s\). When a base pair in the alignment is proposed between columns
\(i\) and \(j\), it has some probability of occurring in each of the \(N\) sequences. If a
sequence \(s\) contains a gap at either of the two positions in the alignment
the probability is 0, otherwise the corresponding entry in \(M^s\) is used.

Given positions \((i,j)\) in the alignment, let \((i',j')\) be the original positions in
the unengaged sequence \(s\). The partition score for a given base pair is then
given as:
\[
P_{i,j} = \frac{1}{N} \sum_{s=1}^{N} M^s(i',j').
\]

The score is thus the mean probability assigned to a base pair by the
partition function. If a base pair is undefined in a number of sequences, the
score is lower due to the lack of evolutionary evidence for that particular
base pair. We use the partition function as implemented in the RNAfold
program from the Vienna package (Hofacker et al., 1994).

### 4 RESULTS AND DISCUSSION

#### 4.1 Information decreases with overall similarity

The performance of the different measures was analyzed using MCC as a
function of threshold values. Graphs showing MCC versus threshold for the
individual datasets as well as graphs for the different identity intervals can be
found in the Supplementary material. The performance of the different measures is summarized
in Figure 1: For each measure, the maximum MCC for all the
thresholds we used is shown for each of the three identity intervals.

This makes comparison of the relative performance both between
measures and between the identity intervals easy.

We also analyzed the performance of the different measures as a
function of the number of sequences in the dataset by using sets
containing from 2 to 20 sequences (see Supplementary material).

All measures performed best with many sequences, which is not
surprising, but the relative rating of each measure did not change
dramatically.

As expected, all covariation based measures perform best on the
low identity datasets with a mean MCC of 0.56. As the overall
similarity of the sequences increase, the methods perform worse.

This is due to the fact that all these measures rely on a signal from
sequence variation. If the sequences are too similar, there is no
signal in the alignment and the measures fail. However, the
drop in performance varies between the different methods. Some
methods are drastically affected (e.g. MI\(^s\) and \(B^s\)), while the standard
mutual information measure is only slightly worse. The best measure is BS with \( \text{MCC} = 0.76 \).

While most of the sequence variation based measures gave MCC scores greater than 0.5 on the low identity interval, only BS exceeded 0.5 on the medium identity interval with MCC = 0.56 (mean MCC of 0.41). The normalized mutual information (MI) is the only measure that shows an increase in MCC from the low to the medium overall identity datasets. A likely explanation is that the normalized measure depends on the MI measure, which is only slightly affected by the increase in sequence identity. At the same time, the normalization is based on the entropy of the alignment, which decreases as the alignment becomes more ‘ordered’ with the larger overall identity. This is the second best measure on this identity interval with MCC = 0.84.

When the overall similarity is further increased to the 80–100% interval all measures perform significantly worse, and the mean MCC drops to 0.12. It is clear that using the information content is futile when the sequences are almost identical. The best method is again BS with MCC = 0.28.

For comparison, the partition function is included in the evaluation. This measure is widely used in RNA structure prediction methods but it is based on a completely different approach. All the other measures use evolutionary information from a sequence alignment, while the partition function uses experimentally obtained energy terms to determine base pairing interactions. Since the partition function does not rely on covariation it is not affected by sequence similarity to the same extent as the other measures. For the purpose of discriminating between true and false base pairs the partition function performs very well on all identity intervals.

It is interesting to see that the partition function also performs better on the low identity datasets. This is not due to covariation of the sequences, but due to the benefits of averaging over a divergent set of structure predictions—an effect that is known from ensemble methods (Krogh and Vedelsby, 1995).

On the low identity datasets the partition function obtains an MCC of 0.92, which is significantly better than the covariation measures. For the 0.60–0.80% interval, the MCC is decreased to 0.88. Finally, the lowest MCC obtained using the partition function (0.68 for the high identity dataset) is only slightly lower than the highest MCC obtained by any other measure (0.76 for BS on the low identity dataset). Based on these results, it is clear that the partition function is an excellent measure for discriminating between true and false base pairs, as it was also shown in Mathews (2004). Combining the partition function with one of the covariation measures makes obvious sense, such as it is done in e.g. RNAalifold (Hofacker et al., 2002).

### 4.2 Extending the basic measure

Since the performance is so dependent on the sequence similarity only the results from the low identity datasets will be discussed in the following. Referring to Figure 1 and Table 2, it can be seen that the standard MI only achieves an MCC of 0.41. The measure is therefore a poor choice for discriminating between true and false base pairs. Normalizing by the entropy (MI\( ^e \)) as suggested by Martin et al. (2005) does not help, on the contrary the performance decreases (MCC = 0.24). A partial explanation for the poor behaviour of these MI scores is that they rely on many frequency estimates, which results in a poor signal-to-noise ratio unless an unrealistically high number of sequences is available.

Constraining the measure to only consider canonical base pairs (MI\( ^b \)) gives an increase in MCC to 0.52, and further using the stacking formulation in MI\( ^p+b,e \) yields MCC = 0.55. This is the expected behaviour since the new formulations should limit the noise and improve the true structural signal. The simple extension of adding a gap penalty (MI\( ^g \)) gives a considerable improvement of the MCC to 0.59. It should be noted that this improvement in MCC can be due to the alignments used: since gaps in stems are actively avoided in the hand-curated alignments, the bonus from the gap penalty is boosted. Nevertheless, since gaps are unwanted in structurally important stretches, this result is promising.

Combining the use of canonical base pairs with a gap penalty (MI\( ^p+b,g,e \)) gives a good MCC of 0.64, which is significantly better than either one alone. Adding a gap penalty to the stacking measure (MI\( ^p+b,e,g \)) increases the MCC to 0.64 compared to stacking alone. The gap penalty thus shows good improvements when used in combination with the different measures. It is possible that the gap penalty could be further optimized, since it is a simple formulation that is used.

### Table 2. Summary of performance

<table>
<thead>
<tr>
<th>Measure</th>
<th>Low identity (( \gamma_\text{MCC} ))</th>
<th>Medium identity (( \gamma_\text{MCC} ))</th>
<th>High identity (( \gamma_\text{MCC} ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI</td>
<td>(1.01,0.41)</td>
<td>(1.01,0.39)</td>
<td>(0.57,0.10)</td>
</tr>
<tr>
<td>MI( ^e )</td>
<td>(0.91,0.59)</td>
<td>(0.88,0.41)</td>
<td>(0.58,0.13)</td>
</tr>
<tr>
<td>MI( ^b )</td>
<td>(0.60,0.52)</td>
<td>(0.52,0.39)</td>
<td>(0.08,0.13)</td>
</tr>
<tr>
<td>MI( ^p )</td>
<td>(0.30,0.24)</td>
<td>(0.46,0.44)</td>
<td>(0.28,0.12)</td>
</tr>
<tr>
<td>MI( ^b,p,e )</td>
<td>(0.68,0.55)</td>
<td>(0.78,0.34)</td>
<td>(0.66,0.03)</td>
</tr>
<tr>
<td>MI( ^p+b,e )</td>
<td>(0.50,0.64)</td>
<td>(0.45,0.38)</td>
<td>(0.77,0.06)</td>
</tr>
<tr>
<td>MI( ^p+b,g,e )</td>
<td>(0.45,0.64)</td>
<td>(0.41,0.40)</td>
<td>(0.09,0.12)</td>
</tr>
<tr>
<td>BS</td>
<td>(0.61,0.66)</td>
<td>(0.85,0.42)</td>
<td>(0.58,0.16)</td>
</tr>
<tr>
<td>BS( ^p+b,e )</td>
<td>(0.36,0.76)</td>
<td>(0.52,0.56)</td>
<td>(0.42,0.28)</td>
</tr>
<tr>
<td>BS( ^p+b,g,e )</td>
<td>(0.22,0.92)</td>
<td>(0.26,0.88)</td>
<td>(0.19,0.68)</td>
</tr>
</tbody>
</table>

The table shows the optimal pairs of threshold (\( \gamma \)) and MCC obtained in this study for each measure used on the three identity intervals.
The measure used in RNAalifold also limits the allowed base pairs and explicitly penalizes gaps and inconsistent pairs. This gives a good MCC of 0.66 which is better than any of the MI based measures. The stacking version of the RNAalifold measure (BS) is even better giving an MCC of 0.76. The RNAalifold measure and its stacking extension are comparable to MI^{P/P} and MI^{P/P,BS} respectively: they all include gap penalties and only focus on canonical base pairs. The main difference is the explicit penalty for inconsistencies in the RNAalifold measure, which gives a significant improvement in MCC.

The advantage of using the symmetric stacking is that the level of noise is smoothed while the signal is only slightly affected (mainly in stem ends), thus improving the signal-to-noise ratio. In parallel with the BS measure, a symmetric version of the MI^{P/P,BS} measure was tested. However, experiments showed that the performance decreased which is possibly due to the decreased signal-to-noise ratio as a result of the increase in variables to estimate (data not shown). We also tested a next-to-nearest-neighbour version of the BS measure which resulted in a slight improvement. However, the weighting scheme was rather ad hoc and could not be justified (the fourth row of Pascal’s triangle; data not shown). It might be worth doing further investigations into this stacking model.

5 CONCLUSION

The standard MI is not well suited for the task of secondary structure prediction, and not all the proposed extensions can remedy this. Adding a simple gap penalty, though, greatly increases the performance as shown above. Likewise, counting only the most common base pairs also gives better results. It is also seen that combining different extensions in general improve the MCC.

The covariation measure used in RNAalifold (Hofacker et al., 2002) performs very well and is a good choice due to its simplicity. The simple extension to include stacking in this measure also shows potential and might be worth exploiting in the future. Of the covariation based measures evaluated in this work, this was the most discriminative. The partition function used for comparison was the most powerful, though. The performance of the individual measures for the three similarity classes is summarized in Table 2 together with the threshold values.

In this study, we have analyzed a number of measures to distinguish true and false base pairs. Standard MI is still widely used, but our evaluation indicates that this is not the best measure. The results presented here should give other researchers an idea of useful information measures for RNA secondary structure prediction and—just as importantly—an idea of which measures are not useful.

ACKNOWLEDGEMENTS

The authors would like to thank Jonathan P. Bollback for discussions regarding inclusion of stacking in mutual information, and Ivo L. Hofacker for information on the RNAalifold covariation measure. The authors also thank four anonymous reviewers for helpful comments, especially reviewer 4 for suggesting the symmetric version of RNAalifold with stacking. S.L. was supported by a Carlsberg Foundation Grant (21-00-0680).

Conflict of Interest: none declared.

REFERENCES

Havgaard,J.H. et al. (2005) Pairwise local structure alignment of RNA sequences with sequence similarity less than 40%. Bioinformatics, 21, 1815–1824.


Chapter 2

MASTR: multiple alignment and structure prediction of non-coding RNAs using simulated annealing

Stinus Lindgreen, Paul P. Gardner and Anders Krogh

Sequence analysis

MASTR: multiple alignment and structure prediction of non-coding RNAs using simulated annealing

Stinus Lindgreen1,*, Paul P. Gardner1,2,† and Anders Krogh1

1Bioinformatics Centre and 2Molecular Evolution Group, Department of Molecular Biology, University of Copenhagen, Ole Maaloes Vej 5, 2200 Copenhagen N, Denmark

Received on August 20, 2007; revised on October 11, 2007; accepted on October 13, 2007
Advance Access publication November 15, 2007

ABSTRACT

Motivation: As more non-coding RNAs are discovered, the importance of methods for RNA analysis increases. Since the structure of ncRNA is intimately tied to the function of the molecule, programs for RNA structure prediction are necessary tools in this growing field of research. Furthermore, it is known that RNA structure is often evolutionarily more conserved than sequence. However, few existing methods are capable of simultaneously considering multiple sequence alignment and structure prediction.

Result: We present a novel solution to the problem of simultaneous structure prediction and multiple alignment of RNA sequences. Using Markov chain Monte Carlo in a simulated annealing framework, the algorithm MASTR (Multiple Alignment of Structural RNAs) iteratively improves both sequence alignment and structure prediction for a set of RNA sequences. This is done by minimizing a combined cost function that considers sequence conservation, covariation and basepairing probabilities. The results show that the method is very competitive to similar programs available today, both in terms of accuracy and computational efficiency.

Availability: Source code available from http://mastr.binf.ku.dk/

Contact: stinus@binf.ku.dk

1 INTRODUCTION

In recent years, the amount of evidence that RNAs play a much more active role in the cell than previously thought has grown dramatically. The view has now shifted away from the assumption that non-coding RNAs (ncRNA) merely helped in the protein synthesis (e.g. tRNA, rRNA), and today a wide variety of catalytically active RNAs or ribozymes have been characterized. It has also become clear that ncRNA is a very diverse group of molecules both in terms of function and structure.

RNA molecules have been found to play important and diverse roles (Athanasius F. Bomfínewerer Consortium et al., 2007; Bomfínewerer et al., 2005; Meyer, 2007); the recently discovered family of microRNAs (miRNA) is involved in gene expression and cell specialization, vault RNAs seem to be involved in multi-drug resistance important for the treatment of cancer, and small nuclear RNAs (snRNA) are key players in the splicing of pre-mRNA. A large number of other ncRNA families have yet to be functionally characterized.

It has also become clear that these non-protein coding genes vary greatly in size, ranging from microRNAs of ~20 nt to more than 10000 nt in RNAs involved in eukaryotic gene silencing (Jossinet et al., 2007), and also that they are transcribed in different ways: some reside in introns of protein coding genes, while others are large transcripts that include introns and the possibility of alternative splicing although they lack the open reading frame of a protein coding gene (Meyer, 2007).

Experimental studies show that a huge fraction of the human genome is transcribed (Cheng et al., 2005), and computational studies show evidence that thousands of structurally conserved RNAs can be found in the human genome (Pedersen et al., 2006; Washietl et al., 2005). There is therefore little doubt that RNAs are biologically very important, and the structural analysis of RNA sequences is a field of growing interest. Through evolution, the sequences of related RNAs can diverge although the structure remains conserved. Pure sequence comparison methods therefore fail when applied to ncRNAs that have diverged too much (Gardner et al., 2005).

It is ultimately the tertiary structure that determines the function of the molecule, and advances are being made in this field (Das and Baker, 2007; Shapiro et al., 2007). However, in the case of RNA it is often sufficient to determine the secondary structure. The reason is that the formation of secondary structure is fast, and the basepairing interactions are strong. The secondary structure, therefore, contributes the major part of the folding energetics, forming a stable scaffold for the formation of tertiary interactions (Ono and Tinoco, 2004).

There exist methods to fold a single RNA sequence either by maximizing basepairing interactions (Nussinov et al., 1978), or by minimizing the free energy of the structure [infold (Zuker and Stiegler, 1981); RNAfold (Hofacker, 2003)]. Another approach is to use an existing sequence alignment and predict a consensus structure based on this. In RNAalifold (Hofacker et al., 2002), this has been pursued using a combination of free energy and covariation. In Pfold (Knudsen and Hein, 1999, 2003), a stochastic context-free grammar (SCFG)
is used to predict a common structure from a multiple alignment. If pseudoknots are disregarded, an RNA structure can be represented as a tree. Since comparison of strings can be extended to trees (Tai, 1979; Zhang and Shasha, 1989), alignments could be based on the structures directly. In RNAforester (Höchsmann et al., 2003), the input is a set of RNA sequences with (possibly predicted) secondary structures, and the problem thus becomes a forest alignment problem. The program performs either local or global alignment of the structures, and the output is an alignment and predicted consensus structure based on the structural similarities. MARNa (Siebert and Backofen, 2005) is another heuristic method, where a multiple structural alignment is inferred from all pairwise alignments of secondary structures. Due to the tight relationship between sequence and structure, the solution to the sequence alignment problem and the structure prediction is interdependent. Whether aligning without considering the structure, or folding without considering sequence alignment, information is ignored. Ideally, one should therefore perform the sequence alignment and structure prediction in parallel. In 1985, Sankoff presented an exact solution to this AP-hard problem (Sankoff, 1985), but the exponential running time of $O(L^{n^2})$ and memory usage of $O(L^{2n})$ makes it intractable even for problems of moderate size. Various implementations of the Sankoff algorithm exist. Foldalign (Gorodkin et al., 1997; Havgaard et al., 2005) and Dynalign (Mathews and Turner, 2002) are both limited to local pairwise alignment. In PMcomp/PMmulti (Hofacker et al., 2004), the optimal alignment of two basepairing probability matrices is found instead of aligning the RNA sequences per se. A multiple alignment can be built using progressive alignment of the basepairing probability matrices. A similar progressive approach is used in FoldalignM (Torarinsson et al., 2007). LocARNa (Wili et al., 2007) is a local alignment tool similar to but more efficient than PMcomp, and this program can also be used to do progressive multiple alignment and structure prediction. In RNAcast (Reeder and Giegerich, 2005), the consensus structure problem is dealt with in a different way: by using abstract shapes (Giegerich et al., 2004), where the structures can be regarded without all details but only using the layout of the structure, the search space is reduced. RNAcast predicts the best common shape for all the sequences and, for each sequence, the energetically best structure.

In RNA Sampler (Xu et al., 2007) stems are the core building blocks. For each sequence, a list of all possible stems consisting of consecutive $A$–$U$, $G$–$C$ and $G$–$U$ pairs is generated. A pairwise alignment is found by aligning all stems from one sequence with all stems from another, and the loop regions are aligned using ClustalW (Thompson et al., 1994). Since bulges are not allowed in stems, the alignment process can be done efficiently by sliding one stem along the other. Such a block of aligned stems has a conservation score including both nucleotide alignment probabilities and basepairing probabilities. From the set of blocks, a common structure is found by sampling, and the probability of a block being chosen depends on the conservation score. The probability matrices are then updated based on the sampled structures, and the process is iterated until convergence. This process has been extended to multiple sequences by considering all pairs in a set of multiple RNA sequences.

SimulFold (Meyer and Miklos, 2007) is a fully probabilistic model using Bayesian Markov chain Monte Carlo. The program takes as input a set of unaligned sequences $D$ and samples both multiple alignment $A$, secondary structure $S$ and a phylogenetic tree $T$ from the joint posterior probability $P(S, A, T | D)$. This very comprehensive program came out very recently, but although it has some methodology in common with MASTR (e.g. sampling based on the likelihood of the solution) it does so in a very different way, which also shows in the computational complexity of the program.

Since the exact solution to the problem is too time and memory consuming to be pursued, all the methods above are simplified in one way or another. Furthermore, it has been suggested that the optimal minimum free energy structure is not necessarily a good solution to the consensus structure problem (Ding and Lawrence, 2003). We therefore pursue a heuristic sampling approach where the structure and sequence alignment can be optimized in parallel. In our approach a cost function (or energy) is defined as a sum of three terms: an alignment term, a structure term and a covariance term. This cost function is minimized using simulated annealing to obtain the combined alignment and structure with minimum cost—the best solution according to the cost function. This optimization is carried out by changing the structure on the basepair level or by moving gaps around in the sequence alignment. The change is then judged by the change in the cost function and either accepted or rejected. The procedure is implemented in the program called MASTR (Multiple Alignment of ST ructural RNAs).

## 2 METHODS

### 2.1 Defining the cost function

To find a solution to the problem of simultaneous multiple alignment and structure prediction, we define a cost function that will be minimized in order to search for the optimal solution. A solution should minimize a combined cost function $\mathcal{L}(A, S)$, which incorporates both the sequence alignment $A$ and the predicted consensus structure $S$. The different parameters used in the program (e.g. scaling parameters and thresholds) have been set using grid optimization. A small number of low identity RNA datasets have been used to optimize the parameters by changing the settings slightly and reevaluating the results. It should be noted that the datasets used for optimizing the parameters are not the same as in the test, and that the datasets do not cover all the families used in the comparison.

#### 2.1.1 Calculating alignment cost

There exist many ways of determining the cost of a multiple alignment: Sum of Pairs using a substitution matrix and minimization of the entropy of the alignment are two well-known examples (Durbin et al., 1998), and using a phylogenetic tree to sum the pairwise alignments inferred by the edges has also been pursued (Hein, 1989).

During the development of the algorithms, various sequence cost functions were examined. Sum of Pairs and different entropy-based measures were tested using both single nucleotide and dinucleotide domains. We selected the best performing cost function, which proved to be a log-likelihood cost function inspired by Hidden Markov models (HMMs) over single nucleotides.
The cost function is fully probabilistic in its treatment of both gaps and nucleotides. We assume independence between the sites in the alignment. When calculating the cost, we have an alignment of length $L$ consisting of $N$ sequences. Let $x'_i$ denote the $i$th character in sequence $i$, and let $P(x'_i)$ denote the probability of seeing character $x'_i$ at this specific position. Assuming the sites are independent, the probability of the multiple alignment $A$ becomes:

\[
P(A) = \prod_{i=1}^{N} \prod_{j=1}^{L} P(x'_i)
\]

The individual character probabilities need to be determined and gaps have to be taken into account. If $x'_i$ is a gap we have two cases: let $P_{GO}$ denote the gap open probability, i.e. the probability of having a gap at position $j$ given that position $j-1$ contained a nucleotide. Similarly, $P_{GE}$ is the gap extension probability used when both position $j$ and $j-1$ contain a gap. Both of these probabilities can be estimated from known structural alignments. In the program, they are set to $P_{GO} = 0.5$ and $P_{GE} = 0.74$.

If $x'_i$ is a nucleotide from the alphabet $\Sigma = \{A,C,G,U\}$, the probability $P(x'_i)$ is calculated based on the nucleotides that comprise the column. Additionally, the probability is dependent on the preceding character. If $x'_{i-1} = \sim$, we have a gap closing, and the probability is multiplied by $(1 - P_{GO})$. Similarly, if $x'_{i-1} \in \Sigma$, the probability is multiplied by $(1 - P_{GE})$. Let $c_i(a)$ be the number of occurrences of nucleotide $a \in \Sigma$ at position $j$ in the alignment. The probabilities are given as:

\[
P(x'_i) = \begin{cases} 
  P_{GO} & x'_i = \sim, x'_{i-1} \in \Sigma \\
  (1 - P_{GO}) \sum_{a \in \Sigma} P_{GO} P_{a} & x'_i = a \in \Sigma, x'_{i-1} \in \Sigma \\
  (1 - P_{GE}) \sum_{a \in \Sigma} P_{GE} P_{a} & x'_i = a \in \Sigma, x'_{i-1} = \sim
\end{cases}
\]

A simple pseudo-count function is used where $c_i(a)$ is incremented by 1 for each $a \in \Sigma$ and $1 \leq j \leq L$. An IUPAC ambiguity character is exchanged with one of the nucleotides it symbolizes with equal probability. For instance, if an $N$ occurs in a sequence, it is replaced by any one of the 4 nt with 25% chance each. Similarly, an $S$ will be exchanged with either a $C$ or a $G$ with a 50% chance each. This exchange is done once in the beginning of the program. Having these probabilities, $P(A)$ can be calculated. The cost function used is the negative log-likelihood based on the alignment probability:

\[
Q(A) = -1/N \log(P(A))
\] (1)

2.1.2 Calculating structure cost

The cost of the structure is defined as the sum of the cost of the individual basepairs. Let $S$ be the structure consisting of basepairs $(i,j)$:

\[
\text{cost}(S) = \sum_{(i,j) \in S} \text{cost}(i,j)
\]

There are two ways to score the structure: by the free energy of single sequences and by covariation. In the present work, we use the two measures that proved best at predicting true basepairs in our previous study (Lindgreen et al., 2006): The McCaskill basepair probabilities (McCaskill, 1990), called $P(b_{ij})$, and a novel version of the covariation measure used in RNAalifold (Hofacker et al., 2002) extended to include stacking of basepairs, called $C(b_{ij})$.

McCaskill showed how to calculate the partition function over all possible secondary structures of an RNA sequence. The basepair probabilities are found using the weighted Boltzman ensemble favoring more stable structures. We use RNAfold, which is part of the Vienna package (Hofacker, 2003), to calculate the probability matrices. Since gaps are added to the sequences as part of the alignment this has to be taken into account when indexing the matrices: the partition function is calculated once for each ungapped sequence $s = 1, …, N$ before the optimization starts, and the results are stored in individual probability matrices $M^s$. These matrices do not change throughout the algorithm. For a basepair $(i,j)$ in the alignment, we need to correct the indices to be able to find the probability for that particular basepair. Let these gap-corrected indices be denoted $(i', j')$, where $i' = i - M^s$ and $M^s$ is the number of gaps preceding position $i$ in sequence $s$, and similarly for index $j$. The probability for the basepair in sequence $s$ is then found as $M^s(i', j')$. If either $i$ or $j$ is a gap in sequence $s$, $M^s(i', j') = 0$. A basepair $(i,j)$ in the alignment is scored by the mean probability:

\[
P(b_{ij}) = 1/N \sum_{s=1}^{N} M^s(i', j')
\]

To transform this into a cost function for the basepairs, the negative logarithm of the mean probability is taken and a threshold is introduced. The threshold reflects the background probability $P_{null}$ of random basepairs found in the probability matrices:

\[
\text{cost}(i,j) = - \log_2(P(b_{ij})) + \log_2(P_{null})
\] (2)

A background probability of $P_{null} = 0.25$ is used based on parameter optimization (data not shown).

Through evolution, related RNA sequences can mutate which leads to different sequences of nucleotides while the same core secondary structure is retained. When a mutation happens at a position that is involved in a basepair, selection favors mutations at the other position that maintain the structure and molecular function. This is known as compensatory mutations. Thus, structure is often more conserved than sequence, and this signal can be measured by a covariation score.

In Lindgreen et al. (2006), we analyzed various measures of covariation. We refer to this article for details, but here the chosen cost function will be briefly explained. The RNAalifold measure uses a matrix $\Pi_{i,j}^s$ for each sequence $s$, where $\Pi_{i,j}^s = 1$ if sequence $s$ can form a basepair between position $i$ and $j$, and $\Pi_{i,j}^s = 0$ otherwise. The function $\delta(x'_i x'_j, \Pi_{i,j}^s)$ measures the Hamming distance between two aligned pairs at positions $i$ and $j$ in sequences $s$ and $t$. The goal is to measure the fraction of consistently aligned pairs. A penalty term, $\phi_{i,j}$, measures the fraction of sequences with inconsistent pairs in the alignment. The covariation is then found as:

\[
R_{i,j} = \frac{1}{N} \sum_s \delta(x'_i x'_j, \Pi_{i,j}^s) - \phi_{i,j}
\]

To add stacking information, the two basepairs enclosing the pair in question are also considered, but more weight is put on the actual pair:

\[
C(b_{ij}) = R_{i,j} + 2 + R_{i-1,j+1}
\]

To turn this into a cost function, the same approach is used as for the partition function above. The covariation score is negated and a threshold value added:

\[
\text{cost}_{(i,j)} = - C(b_{ij}) + \phi
\] (3)

A threshold of $\phi = 0.25$ is used. Using the two cost functions $\text{cost}_{(i,j)}$ and $\text{cost}_{(i,j)}$ [Equations (2) and (3), respectively], the predicted structure can be evaluated and a move either accepted or rejected based on Equation (4) below.

2.1.3 Combined cost

Since we simultaneously optimize both sequence alignment and structure prediction, the cost function is a combination of three terms: the log-likelihood cost in Equation (1), the basepair probability cost in Equation (2), and the covariation cost
in Equation (3). The cost of the secondary structure is given as a sum over all basepairs in the structure $S$:

$$\text{cost}(A, S) = Q(A) + \sum_{i,j} (a \cdot \text{cost}(i,j) + \beta \cdot \text{cost}(i,j))$$

The parameters $a$ and $\beta$ are parameters used to balance the contribution from the different terms in the combined cost. As default, they are set to $a = 1.5$ and $\beta = 0.6$, which are obtained from an initial grid search parameter optimization (data not shown).

### 2.2 Optimizing the solution

Simulated annealing (Kirkpatrick et al., 1983) is an optimization technique inspired by the physical process of annealing, which describes the slow cooling of material to form a crystal structure. The idea is that the positions of the individual atoms can be described as a probability distribution depending on the temperature of the system: at high temperatures the atoms have a high energy and therefore move around, but as the temperature is lowered, the system becomes more stable. The goal is to form crystals with few defects, and the most stable crystal structure is the one with the lowest free energy. If the temperature of the system is decreased too fast, the crystal structure becomes brittle since the system becomes stuck in a local energy minimum. If the temperature is decreased slowly, the local energy minima can be avoided due to the thermal fluctuations, and the structure becomes more ordered and stable, and the minimum free energy conformation may be reached.

Simulated annealing works in analogy to this. In order to escape from local minima of a cost or energy function, steps towards worse states (i.e. higher cost) should be taken often in the beginning (at high temperature) and occasionally later at lower temperatures. This is done in a Monte Carlo simulation with an artificial temperature parameter that has absolutely no physical meaning. The probability of acceptance depends on the change in cost (huge increases should be accordingly improbable) as well as on the number of iterations (since the system is closer to the ‘stable’ optimum towards the end). Given an infinite amount of time, it can be shown that simulated annealing will approach the optimal solution to any finite problem (Häggestrom, 2002). Simulated annealing can be used to minimize any cost function, and has for instance been used for multiple alignment (Lukashin et al., 1992).

Simulated annealing depends on an artificial temperature $T$ that decreases over time. Initially the temperature should be high enough to give an ‘unstable system’—in this case an alignment prone to changes. As more and more changes are sampled, the temperature decreases to ‘stabilize’ the system. Normally the temperature decreases exponentially (Lukashin et al., 1992), although there is no theoretical reason for this. If the new cost is lower than the previous, the change is always accepted. If the change increases the cost, the chance of acceptance $P$ depends on the old cost $C_{\text{OLD}}$ and the new (larger) cost $C_{\text{NEW}}$ and the temperature $T$:

$$P = \exp\left(\frac{C_{\text{NEW}} - C_{\text{OLD}}}{T}\right)$$

This is known as the Metropolis–Hastings algorithm (Hastings, 1970, Kirkpatrick et al., 1983). Using this, the possible states are sampled based on the cost of the current state. Since a state only depends on the previous state, this generates a Markov chain. In combination, MCMC using simulated annealing can be used to sample the solution space of multiple alignments and RNA structures. Changes can be made by moving the gaps in the alignment and by adding or removing basepairs in the structure, and the move is either rejected or accepted based on the change in cost.

The initial alignment is built by adding gaps at random to all sequences until they have equal length. By default, the length of the initial alignment is $1.06 \cdot L_{\text{max}}$ where $L_{\text{max}}$ is the length of the longest sequence. The moves through the solution space can either affect the sequence alignment or the structure. Since it makes little sense to try and deduce a common structure from randomly aligned sequences, the first iterations are purely sequence moves. As the alignment becomes more stable, we start doing a combination of sequence and structure moves.

The total number of iterations performed depends both on the length of the longest sequence, since that affects the number of structure moves needed, and on the size of the alignment, since that affects the number of sequence moves needed. The alignment size is measured as the total number of nucleotides in the dataset, $N_{\text{total}}$. The dependencies are denoted $N_{\text{dep}}$ and $L_{\text{dep}}$, respectively, and the number of iterations is found as:

$$I = N_{\text{dep}} \cdot N_{\text{total}} + L_{\text{dep}} \cdot L_{\text{max}}$$

We use $N_{\text{dep}} = 1000$ and $L_{\text{dep}} = 1700$ as default. After initially only performing sequence moves, a mixture of alignment and structure altering moves are performed. The structure moves are initiated either after a fixed fraction of the total number of iterations or, as per default, after $N_{\text{dep}} \cdot N_{\text{total}}$ iterations. The remaining iterations are a mix of sequence and structure moves. The ratio between the two is set by a parameter $R$. Per default, $R = 0.75$ of the last iterations are structure moves.

Initially, all moves are accepted (i.e. a temperature of infinity is used) and the first 0.1% of the iterations are used to determine a good starting temperature. These results are used to estimate the standard deviation $\sigma$ of the cost distribution. By deciding on the desired initial probability of acceptance $P_0$ the temperature $T_0$ can be determined:

$$T_0 = \frac{\sigma}{\log(P_0)}$$

We use $P_0 = 0.99$ as default. The scaling of the temperature has to make sure that we end up close to $T = 0$. An exponential scaling is used:

$$T_{\text{new}} = T_{\text{old}} \cdot \tau \quad \text{where} \quad 0 < \tau < 1$$

Wanting the final temperature to be $T_{\text{final}} = 10^{-5}$, this yields:

$$\tau = \exp\left(\frac{\log\left(\frac{1}{10^5}\right)}{l}\right)$$

#### 2.2.1 Sequence moves

The moves aimed at changing the sequence alignment do this by moving gaps in the sequences. They can of course be moved without altering the order of the nucleotides. Three different types of moves are implemented which in combination ensures that the alignment can be reduced, extended and altered locally.

- **Gap block move**: local changes are facilitated through gap blocks. A gap block is a subsequence consisting of 1 or more consecutive gaps in 1 or more aligned sequences. To make this move, a random gap in a random sequence is picked. Then the gap block is extended vertically with probability 0.85 through the other sequences containing a gap at that position. Afterwards, the gap block is extended horizontally to both sides with probability 0.85 if all the chosen sequences contain a gap there. Finally, the gap block is moved to a randomly chosen new position in the alignment. The procedure is illustrated in Figure 1 and constitutes 85% of the sequence moves.

- **Gap insertion**: insertion of gaps has to insert the same number of gaps in all sequences. One could insert the gaps at random positions in all sequences, but that would greatly affect the cost of the alignment. Instead, the gaps are inserted in either end of the alignment. From these positions the gaps can diffuse into the alignment as needed. These moves constitute 10% of the sequence moves.
2.2.2 Structure moves. Structural moves either add or delete basepairs. The structure is forced to non-crossing basepairs (i.e. prediction of pseudoknots is not yet supported), and a minimum loop length of 3 nt is ensured. Using the three simple moves described below, the structure can be built, extended and reduced.

- Adding a basepair: a new basepair is added by choosing a nucleotide pair \((i,j)\) at random and adding it to the structure if it does not violate the constraints. These moves constitute 70% of the structure moves.

- Extending a stem: a stem is extended by choosing a basepair \((i,j)\) already in the structure. The stem that includes basepair \((i,j)\) is then extended by adding a new basepair to it, with a 50% chance of extending the stem either internally or externally. These moves constitute 20% of the structure moves.

- Deleting a basepair: deleting a basepair is done by choosing a pair \((i,j)\) in the structure and removing it. This cannot lead to any new violations of the constraints. These moves constitute 10% of the structure moves.

2.3 Datasets

Since consensus structures were not available from BRalibase II (Gardner et al., 2005) at the time, we sampled alignments with consensus structures in much the same way as in the BRalibase study. MASTR relies on the partition function to calculate basepair probability matrices, so we have chosen to use only short (~70–250 nt) sequences in the test. There are known problems when using the partition function to calculate basepair probability matrices for long sequences. Hence, the program will not perform well on long sequences until this has been dealt with.

Datasets were generated from large, trusted seed alignments obtained from Rfam (Griffiths-Jones et al., 2005). The 5 families chosen are tRNA, 3S rRNA and U5, 8 families from TPP and 2 datasets from IRES. In total, 52 datasets were sampled and details on average pairwise identity, number of sequences and average sequence lengths can be seen in Table 1. The datasets can be obtained from http://mastr.binf.ku.dk/.

<table>
<thead>
<tr>
<th>Family</th>
<th>ID</th>
<th>Seqs</th>
<th>Length</th>
<th>MASTR MCC</th>
<th>SPS MCC</th>
<th>RNA Sampler MCC</th>
<th>SPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>tRNA</td>
<td>35</td>
<td>11</td>
<td>74</td>
<td>0.72</td>
<td>0.59</td>
<td>0.90</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>11</td>
<td>72</td>
<td>0.89</td>
<td>0.84</td>
<td>0.97</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>12</td>
<td>71</td>
<td>0.89</td>
<td>0.88</td>
<td>0.94</td>
<td>0.88</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>12</td>
<td>73</td>
<td>0.99</td>
<td>0.97</td>
<td>0.85</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>12</td>
<td>73</td>
<td>1.00</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>11</td>
<td>73</td>
<td>1.00</td>
<td>0.99</td>
<td>1.00</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>11</td>
<td>73</td>
<td>1.00</td>
<td>0.99</td>
<td>1.00</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>68</td>
<td>7</td>
<td>73</td>
<td>0.89</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>5</td>
<td>72</td>
<td>1.00</td>
<td>0.98</td>
<td>1.00</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>6</td>
<td>72</td>
<td>1.00</td>
<td>0.99</td>
<td>1.00</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>84</td>
<td>6</td>
<td>69</td>
<td>0.14</td>
<td>0.98</td>
<td>0.31</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>7</td>
<td>72</td>
<td>0.83</td>
<td>1.00</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>8</td>
<td>73</td>
<td>0.97</td>
<td>1.00</td>
<td>0.97</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>97</td>
<td>6</td>
<td>73</td>
<td>0.34</td>
<td>1.00</td>
<td>0.48</td>
<td>1.00</td>
</tr>
<tr>
<td>3S rRNA</td>
<td>36</td>
<td>5</td>
<td>110</td>
<td>0.45</td>
<td>0.62</td>
<td>0.51</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>6</td>
<td>111</td>
<td>0.66</td>
<td>0.74</td>
<td>0.62</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>8</td>
<td>116</td>
<td>0.54</td>
<td>0.76</td>
<td>0.41</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>9</td>
<td>113</td>
<td>0.59</td>
<td>0.74</td>
<td>0.52</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>10</td>
<td>113</td>
<td>0.56</td>
<td>0.91</td>
<td>0.40</td>
<td>0.94</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>11</td>
<td>118</td>
<td>0.89</td>
<td>0.96</td>
<td>0.75</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>11</td>
<td>117</td>
<td>0.83</td>
<td>0.97</td>
<td>0.81</td>
<td>0.97</td>
</tr>
<tr>
<td></td>
<td>69</td>
<td>10</td>
<td>116</td>
<td>0.52</td>
<td>0.95</td>
<td>0.50</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>9</td>
<td>115</td>
<td>0.58</td>
<td>0.96</td>
<td>0.53</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>11</td>
<td>118</td>
<td>0.93</td>
<td>1.00</td>
<td>0.85</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>15</td>
<td>117</td>
<td>0.11</td>
<td>1.00</td>
<td>0.13</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>89</td>
<td>15</td>
<td>117</td>
<td>−0.04</td>
<td>0.97</td>
<td>0.01</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>20</td>
<td>117</td>
<td>0.20</td>
<td>1.00</td>
<td>0.12</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>97</td>
<td>13</td>
<td>117</td>
<td>0.38</td>
<td>1.00</td>
<td>0.10</td>
<td>0.98</td>
</tr>
<tr>
<td>U5</td>
<td>36</td>
<td>5</td>
<td>122</td>
<td>0.40</td>
<td>0.33</td>
<td>0.44</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>6</td>
<td>123</td>
<td>0.48</td>
<td>0.37</td>
<td>0.49</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>9</td>
<td>126</td>
<td>0.46</td>
<td>0.45</td>
<td>0.64</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>8</td>
<td>120</td>
<td>0.50</td>
<td>0.51</td>
<td>0.63</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>10</td>
<td>118</td>
<td>0.67</td>
<td>0.61</td>
<td>0.72</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>9</td>
<td>115</td>
<td>0.71</td>
<td>0.68</td>
<td>0.82</td>
<td>0.65</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>9</td>
<td>114</td>
<td>0.78</td>
<td>0.76</td>
<td>0.52</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>10</td>
<td>115</td>
<td>0.87</td>
<td>0.81</td>
<td>0.93</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>8</td>
<td>116</td>
<td>0.90</td>
<td>0.82</td>
<td>0.93</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>11</td>
<td>119</td>
<td>0.50</td>
<td>0.83</td>
<td>0.24</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>86</td>
<td>7</td>
<td>116</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.94</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>9</td>
<td>115</td>
<td>0.95</td>
<td>0.98</td>
<td>0.92</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>17</td>
<td>115</td>
<td>1.00</td>
<td>1.00</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>97</td>
<td>7</td>
<td>116</td>
<td>0.96</td>
<td>1.00</td>
<td>0.95</td>
<td>1.00</td>
</tr>
<tr>
<td>TPP</td>
<td>35</td>
<td>8</td>
<td>122</td>
<td>0.33</td>
<td>0.42</td>
<td>0.45</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>9</td>
<td>104</td>
<td>0.37</td>
<td>0.65</td>
<td>0.71</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>11</td>
<td>105</td>
<td>0.43</td>
<td>0.75</td>
<td>0.61</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>11</td>
<td>103</td>
<td>0.55</td>
<td>0.72</td>
<td>0.65</td>
<td>0.72</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>11</td>
<td>107</td>
<td>0.70</td>
<td>0.86</td>
<td>0.74</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>58</td>
<td>7</td>
<td>102</td>
<td>0.63</td>
<td>0.91</td>
<td>0.69</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>61</td>
<td>4</td>
<td>101</td>
<td>0.66</td>
<td>0.97</td>
<td>0.59</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>4</td>
<td>90</td>
<td>0.45</td>
<td>0.86</td>
<td>0.33</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>73</td>
<td>4</td>
<td>250</td>
<td>0.54</td>
<td>0.91</td>
<td>0.45</td>
<td>0.86</td>
</tr>
</tbody>
</table>

For each dataset, the average pairwise identity (ID) is shown along with the number of sequences (Seqs) and the average sequence length (Length). The results for MASTR and RNA Sampler are detailed (MCC: structure quality, SPS: alignment quality). For each dataset, the best results are highlighted with bold, and identical results with italics.
3 RESULTS

The program MASTR is implemented in C++ and tested against the programs FoldalignM, LocARNA and RNA Sampler. RNAcast is used to produce input to RNAforester, and Clustal alignments were used as input to RNAalifold. All programs were used with their default settings except for RNAforester where the clustering cutoff had to be changed in order to produce complete alignments of all sequences. FoldalignM does not predict a single consensus structure but returns a structure for each sequence in the final alignment. Therefore, we define the consensus structure to be the basepairs that are predicted for all sequences. The other programs all predict a single multiple alignment and consensus structure.

To evaluate the predicted structures, Matthew’s Correlation Coefficient (MCC) is used:

\[ \text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \]

Let TP be the number of truly predicted basepairs, FP be the number of predicted basepairs not in the reference structure and FN be the number of basepairs in the reference structure not predicted by the program. TN is defined here as the number of possible basepairing interactions in a sequence that are not predicted and not in the reference structure, i.e. pairs of nucleotide \( xy \) that are at least 4 nt apart, and where \( x \in \{AU, UA, CG, GC, UG, GU\} \).

To evaluate the quality of the alignment, the Sum of Pairs score (SPS) (Thompson et al., 1999) is used. SPS is a sensitivity-like measure that compares the predicted alignment to a reference. For each pair of aligned sequences, the number of aligned positions that are present in both the prediction and the reference alignment is counted. The total number of correctly aligned positions is then compared to the total number of aligned non-gap pairs present in the reference alignment. This yields a number between 0 and 1 where 1 is perfect correspondence between prediction and reference.

In Figure 2, the performance of the programs are compared in terms of structure quality (plot a), alignment quality (plot b) and running time (plot c) as a function of the average pairwise identity of the datasets (%ID). The plots are averages over the different RNA families used for each %ID point.

The test shows that MASTR can predict consensus structures of a quality comparable to other existing methods. On the lowest identity datasets MASTR is outperformed by RNA Sampler, but after ~45% ID the structure predictions of MASTR are on average better than or comparable to the best programs tested. MASTR is consistently better than or comparable to all other methods regarding alignment quality. As it can be seen, MASTR is clearly faster than FoldalignM by up to an order of magnitude, while LocARNA is even faster. Clustal + RNAalifold is of course by far the fastest method used. RNAforester has a running time comparable to LocARNA but produces consistently worse alignments—probably due to the fact that all sequences had to be included in the same alignment for comparison. The structure predictions are comparable to FoldalignM.

The dip in the quality of the structure predictions that is visible for all methods in the highest identity range could be explained by the lack of covariation in these datasets. Most of the methods rely on some signal from compensatory mutations, and this signal diminishes as the sequences become too similar. RNA Sampler does not depend on a covariation signal, and this signal diminishes as the sequences become too similar. RNA Sampler is of course by far the fastest.
Likewise, FoldalignM shows an almost monotone increase in the predictions as a function of identity. Since RNA Sampler seems to be the best of the other methods tested here, a more detailed comparison of MASTR and RNA Sampler can be seen in Table 1. In total, 52 datasets are used. The running time is on the same scale for the two programs, although MASTR is in general slightly faster. The structure predictions are better than or equal to RNA Sampler in 28 cases (54%), and the alignments are better than or equal to RNA Sampler in 43 cases (83%). The differences seem to depend both on the RNA family and on the level of identity.

4 DISCUSSION

We have developed a new algorithm for simultaneous alignment and structure prediction of multiple non-coding RNA sequences. As shown above, MASTR is highly competitive both in terms of structure prediction quality, sequence alignment and running time. The program can also handle larger datasets than, e.g. RNA Sampler or FoldalignM.

Although we have not used it in the above tests, it is also possible to add structural constraints if some knowledge is available about one of the sequences (e.g. known basepairs, knowledge about upstream or downstream interactions, or knowledge about non-basepaired positions). Additionally, already aligned sequences can be used as input with or without a consensum structure.

As the testing of the program showed, MASTR does not have top performance on very dissimilar sequences. In this range, one would assume that covariance is important, and it is therefore interesting that RNA sampler, which does not use covariance, is better. One possible explanation for this is that covariance in itself is not enough to deduce structure from alignment. Covariation is only an indicator of conserved basepairs, but it is not sufficient to predict pairing columns [this corresponds well with our previous study (Lindgreen et al., 2006)]. MASTR builds up the structure in small steps, which might make it vulnerable to erroneously high covariation, whereas RNA Sampler makes sure that the alignment is structurally sound by fixing whole stems. MASTR therefore needs to have a relatively stable (and correct) alignment before predicting structure. This could explain the relatively poor performance on low identity datasets, and this should be explored further.

In future work, we would like to make a local version of the program. This would be ideal for dealing with long sequences where there are known problems with the standard basepair probability matrices. Since MASTR does not have the same limitations towards crossing basepairs as pure energy-based methods, an extension to include the prediction of pseudoknots will also be pursued.

One of the advantages of MASTR is that the optimization is decoupled from the cost function, which makes it very easy to change the latter. It also makes it reasonably straightforward to add phylogenetic prediction to the program, which would be similar to the goal of SimulFold (Meyer and Miklos, 2007), but MASTR functions in a very different way. We would also like to make it possible to input a set of related and already aligned sequences together with the set of unaligned sequences. Thus, new sequences can be aligned to reference alignments in a structurally sound manner.

ACKNOWLEDGEMENTS

S.L. acknowledges support from the Novo Scholarship Program. The work was supported by the Carlsberg Foundation and the Danish National Research Foundation. The authors thank two anonymous referees for their helpful comments; especially one reviewer for interesting comments on covariance.

Conflict of Interest: none declared.

REFERENCES

Havgaard,J.H. et al. (2005) Pairwise local structure alignment of RNA sequences with sequence similarity less than 40%. Bioinformatics, 21, 1815–1824.
Hein,J. (1989) A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences when the phylogeny is given. Mol. Biol. Evol., 6, 469–469.


Chapter 3

WAR: Webserver for aligning structural RNAs

Elfar Torarinsson and Stinus Lindgreen

WAR: Webserver for aligning structural RNAs
Elfar Torarinsson1 and Stinus Lindgreen2,*

1Division of Genetics and Bioinformatics, IBHV, Faculty of Life Sciences, University of Copenhagen, Grynneegaardsvej 3, DK-1870 Frederiksberg C and 2Center for Bioinformatics, Department of Molecular Biology, University of Copenhagen, Ole Maaloes Vej 5, DK-2200 Copenhagen N, Denmark

Received January 14, 2008; Revised April 7, 2008; Accepted April 26, 2008

ABSTRACT
We present an easy-to-use webserver that makes it possible to simultaneously use a number of state of the art methods for performing multiple alignment and secondary structure prediction for noncoding RNA sequences. This makes it possible to use the programs without having to download the code and get the programs to run. The results of all the programs are presented on a webpage and can easily be downloaded for further analysis. Additional measures are calculated for each program to make it easier to judge the individual predictions, and a consensus prediction taking all the programs into account is also calculated. This website is free and open to all users and there is no login requirement. The webserver can be found at: http://genome.ku.dk/resources/war.

INTRODUCTION
Over the past few years, different studies have shown how noncoding RNAs (ncRNA) are involved in gene expression, cell specialization, multi-drug resistance, splicing etc. in all living cells (1,2). For instance, only a small part of mammalian genomes encodes protein coding genes, but experiments have shown that a large fraction of the genomes is transcribed (3). Thus, there is potential for a large number of ncRNA transcripts, and there is computational evidence for thousands of structured RNAs in several vertebrate genomes (4–6).

This has given rise to an increased interest in ncRNAs, and since the structure of these molecules is tightly linked to their function, structure prediction methods have received much attention. Many different methods have been developed and they vary greatly in their approaches to the problem. Previously, methods for folding a single sequence by predicting the minimum free energy conformation were pursued [mfold (7), RNAfold (8)]. Today, comparative methods are the norm where a multiple alignment of a set of related RNA sequences is a part of the approach. Either the alignment is predicted alongside the consensus structure [i.e. the Sankoff-approach (9)] or an alignment is part of the input to the structure prediction [e.g. RNAalifold (10)].

The benefit of using comparative methods is that more information is available than for single sequence approaches. Many programs have been published over the past few years, and it can be difficult for a user to determine which one to use, to judge the different predictions, and sometimes even to run the programs. Since the performance of the different programs depend on many factors such as sequence length and identity, a specific program will not always perform best. Using an ensemble of programs therefore makes it easier to get a good idea of the correct result.

Here, we present an easy way to run a selection of methods and get a combined view of the predictions. The user simply inputs the sequences to be analyzed, and a selection of programs is automatically run on the dataset. The predictions are analyzed in various ways to make the output more informative for the user. The results are presented on a webpage, where one can easily download the different predictions and compare the relative performance of the individual programs. We also present a combined consensus prediction based on the results.

MATERIALS AND METHODS
The webserver for aligning structural RNAs (WAR) performs multiple alignment and secondary structure prediction on a dataset using a number of programs. The input to the server is the RNA sequences to be analyzed in Fasta format. The sequences can either be uploaded as a file or copy–pasted to a field on the webpage. The methods chosen are (in alphabetical order):

CMfinder (11): an algorithm based on expectation maximization using covariance models. Searches for RNA motifs combined with structure prediction based on both folding energy and covariation.

*To whom correspondence should be addressed. Tel: +45 35321287; Fax: +45 3532 1281; Email: stinus@binfo.ku.dk
The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors

© 2008 The Author(s)
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
FoldalignM (12): based on the Sankoff approach, where the partition function (13) is used to calculate basepair probability matrices for each sequence. These matrices are then aligned using progressive alignment to produce a multiple alignment and predicted consensus structure. The approach is similar to PMcomp/PMmulti (14).

LaRA (15): a mathematical approach using Lagrangian transformed relaxation. The problem of optimizing alignment and structure is formulated as an integer programming problem, and a numerical optimization approach is used.

MASTR (16): a sampling approach using Markov chain Monte Carlo in a simulated annealing framework, where both structure and alignment is optimized by making small local changes. The score combines the log-likelihood of the alignment, a covariation term and the basepair probabilities.

RNAfold (10) + ClustalW (17): ClustalW is one of the most widely used alignment programs. It performs progressive alignment using a simple guide tree. RNAfold predicts the structure given in an alignment using both the free energy and a covariation measure to evaluate the basepairing regions.

RNAforester (18) + RNAcast (19): RNAforester performs multiple alignment based on an input set of sequences with secondary structures. The output is thus based on structural similarities. The input set is generated using RNAcast that predicts the common structure for each sequence.

RNASampler (20): Possible stems are found for each sequence and the stems are then aligned by comparing all pairs of sequences. A conservation score considering both structure and sequence alignment measures the quality, and a structural alignment is built. Unpaired regions are aligned using ClustalW.

All the programs perform only global alignment, except for CMfinder which is capable of performing local multiple alignment. Optionally, one can use the webserver to perform local alignment. This is done by extracting the best scoring local motif predicted by CMfinder. Then all the selected programs in the webserver are run globally, as usual, on the local region selected by CMfinder.

All the programs are run using their default settings. If the user wants to try other parameter settings, we encourage the use of the webserver and/or source code of the individual programs. There are a few limitations on the use of WAR: the user must submit at least two sequences (note that CMfinder does not work with less than three sequences), no more than 50 sequences can be submitted in one job and a sequence can be no more than 250 nucleotides in length unless the local alignment box is checked; in that case, there is no length limit.

Postprocessing of the results

For each method, the result is presented and can be easily downloaded for further use. The multiple alignment is colored using the coloralin-script from the Vienna package (8) and is shown with a barplot visualizing the conservation for each column. The consensus structure is written on top of the alignment, and the predicted basepairs are color coded to highlight canonical basepairs (i.e. Watson–Crick interactions and GU-basepairs) and compensatory mutations.

The consensus sequence is shown along with the consensus structure as predicted by the program (both as dot-bracket and a Postscript-file). The consensus sequence is defined using the whole range of IUPAC ambiguity characters (21) and is similar to the most informative sequence (22).

To quantify the quality of the alignment, the average pairwise identity is calculated. When a reference alignment is not known, it is not a trivial task to measure the correctness of an alignment. This is one of the reasons why so many different alignment algorithms exist. The identity measure used in the WAR server is not as such a measure of correctness, but instead a measure for the similarity of the alignment. For instance, the correct alignment of highly diverged sequences will by necessity have a lower overall identity than the correct alignment of closely related sequences. However, in the current setting, a number of different methods have all been used on the same dataset and one can get an idea of the quality by comparing the pairwise identity of the alignments. If one program obtains a comparatively low pairwise identity, the alignment is probably worse. The average pairwise identity is calculated by making all the pairwise comparisons between sequences in the multiple alignment and counting the number of aligned positions that are identical. The average fraction of identities is then reported for the whole alignment.

To estimate the thermodynamical stability of the predicted structure, the average free energy is used. The average free energy in itself cannot be used directly as a quality measure, but by comparing the average of the different predictions the relative performance of each can be assessed. For a given prediction, we map the consensus structure to each sequence in the alignment after removing the gaps. The sequence is then folded into this specified structure and the free energy calculated using RNAeval (8). The average free energy for the entire alignment given the predicted consensus structure is then reported.

To evaluate how well the sequence alignment supports the predicted structure, we calculate a covariation score for each basepair, given the alignment. The proposed structure will pair up columns in the alignment, and covariation measures the amount of evidence for a basepair. This is done by calculating how often a variation in one column leads to a variation in the other. We use the measure that proved to be best in a recent study comparing different covariation measures (23) and report the average covariation score for all basepairs. Note that this measure can be negative (due to a penalty term) and greater than 1.

The quality of the predicted structure is also measured as the average basepair probability. For each basepair, we calculate the basepair probability matrix using RNAfold (8,13). For a single sequence, we can then find the probability of each proposed basepair by simply looking in the matrix. For two pairing columns in the alignment, the probability of a basepair is then found as the average probability for that particular basepair in each sequence.
The average probability for the whole structure is then reported. All these results are reported on a single webpage that makes it easy to compare the different methods for both similarities and differences. Alignments and structures are easily downloaded in different formats.

A consensus prediction is made based on all the programs in the following way: T-Coffee (24) is a program that performs multiple alignment using a library of all both local and global pairwise alignments in the set. The library is extended by realigning to a third sequence and weights are calculated based on consistencies within the library. The multiple alignment is then performed progressively based on these weights.

This method can also predict a consensus multiple alignment by building the library from a number of multiple alignments instead. A single consensus multiple alignment is constructed by giving the alignments from all the programs as input to T-Coffee. The consensus structure is found by taking each ungapped sequence and mapping the predicted structure from each program onto it. If a basepair is predicted by at least 50% of the programs, it becomes a part of the consensus structure for that sequence. This gives us a consensus structure based on all the programs for each sequence in the alignment. Each sequence is then aligned to the T-Coffee alignment, making it possible to compare the consensus structure for each sequence, and the basepairs that are present in at least 50% of the sequences become part of the consensus structure for the whole alignment.

WEBSERVER

We illustrate the use of WAR in the following and especially show how the consensus prediction can be useful. For this purpose, we use a tRNA dataset consisting of 10 randomly chosen sequences, but the procedure is the same for any dataset.

If it is known, a reference alignment and structure can be uploaded along with the unaligned sequences. This makes it possible to compare the predictions to the correct answer and thus rank the methods. We use the following scores: the predicted alignment is compared to the reference alignment using the sum of pairs score (SPS), which is a sensitivity-like measure (25). It is based on the fraction of nucleotide pairs aligned in the prediction that are also present in the reference and yields a number between 0 and 1, where 1 is perfect prediction. The predicted structure is compared to the reference using Matthew’s correlation coefficient (MCC), which shows the balance between sensitivity (SEN, i.e. the fraction of correct basepairs that are recovered by the method) and positive predictive value (PPV, i.e. the fraction of predicted basepairs that are also in the reference). MCC lies between −1 and 1, where 1 is perfect prediction.

Using WAR is simple: on the input form, you have to specify the input sequences either in the box or as a file. You also have to input a valid email address to receive notification when the job is complete (note that the email is used only for this notification). There are also some optional settings: you can choose to perform local alignment, you can specify a reference alignment file if available, you can name the submission and you can choose only to run a selection of the programs.

When the programs have finished, a table is shown that summarizes the predictions (Figure 1). If a reference alignment was given, the four rightmost columns summarize the performance (SPS, SEN, PPV and MCC). Otherwise, only the first six columns are shown (program, CPU time, average sequence identity, average free energy, covariation and average basepair probability). These measures are calculated as described earlier.

In the current example, LaRA and RNASampler gave the most correct structure predictions with $MCC = 0.95$. This corresponds with the two programs having the two lowest average free energies ($−17.94$ and $−18.75$, respectively) and the two highest average covariations ($0.97$ and $1.06$, respectively). The average basepair probabilities are also high in both cases ($0.55$ and $0.58$, respectively). The most correct alignment was also predicted by LaRA with $SPS = 0.88$. The measure of pairwise identity in the prediction is in this case $0.40$, which is only the second largest. Notice, however, that the program with the highest pairwise identity also shows a very good $SPS$ of $0.86$.

<table>
<thead>
<tr>
<th>Performance Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Program</strong></td>
</tr>
<tr>
<td>Consensus</td>
</tr>
<tr>
<td>CMFinder</td>
</tr>
<tr>
<td>FoldalignM</td>
</tr>
<tr>
<td>LaRA</td>
</tr>
<tr>
<td>MASTR</td>
</tr>
<tr>
<td>RNAfold</td>
</tr>
<tr>
<td>RNAsforester</td>
</tr>
<tr>
<td>RNASampler</td>
</tr>
</tbody>
</table>

Figure 1. The table showing the results from the different programs.
When looking at the consensus prediction, it is evident that combining the structure predictions from all seven programs yields an improvement in both SEN, PPV and MCC to 1.0, which is better than the previous best prediction with $MCC = 0.95$. Also, the alignment is improved to $SPS = 0.88$, which is better than the single best prediction with $SPS = 0.88$. Looking at the other measures, the average free energy (-19.34) and covariance (1.06) are also best for the consensus, with the pairwise sequence identity (0.41) and basepair probability (0.57) being second best. In this case, using the consensus prediction based on the seven programs is clearly an improvement to using any single program.

Of course, that is not always the case, especially when only a few of the programs make reasonable predictions. Therefore, it is possible to update the consensus (see below) using only selected programs and not, as per default, all of them. For example, if some of the programs predict a very unstable consensus structure with a high free energy, it might be a good idea to update the consensus by removing those programs that perform poorly.

Clicking on any of the links in the first column shows a detailed description of the prediction from the chosen program. These pages are similar for all programs, only the consensus link differs. On the consensus page, the alignment is shown as a heat map at the top illustrating the confidence of the different parts of the alignment (from blue being low to red being high, see Figure 2). If necessary, one can choose which program to include in the consensus and update the alignment and structure. If some of the predictions are very different from the rest, the consensus might be improved by excluding these from the calculation.

Further down the page is additional information, which is calculated for each program (Figure 3). The measures such as average pairwise identity and—if a reference was uploaded—MCC etc. are shown. Furthermore, the predicted consensus structure is shown with the calculated consensus sequence. By pointing with the mouse at the small image-icon, the structure is shown in a small pop-up window and by clicking it is downloaded as a Postscript-file. Below the structure a color-coded image of the alignment is shown, which can be downloaded as a gif-file or a Postscript-file. The raw output data from the program can be downloaded in various formats (Fasta, Clustal, Stockholm and col) in the top right corner of the page.

The consensus prediction is particularly useful when studying unknown structural RNAs. With no prior knowledge, it is hard to judge if getting a good prediction from a single program is reliable. But if all or several of the programs agree, and the heatmap shows a reliable consensus alignment and structure, you can have higher confidence in the results.

**CONCLUSION**

The WAR webserver makes it easy to use a number of methods for aligning and predicting the secondary structure for a set of structural RNAs. With all the focus on ncRNAs, this is a very useful tool for any researcher who wants to analyze sequences, but does not...
want to get the different programs to run him/herself. In time, WAR will be extended with new methods that perform well, and other combinations of alignment tools and structure prediction tools might be pursued.

What makes WAR especially useful is the postprocessing. The different measures calculated as part of the pipeline make it easier for the user to judge the quality of both the alignment and predicted structure, as well as compare the different methods. The calculated consensus alignment and structure is a valuable indicator of the quality of the predictions, especially for new, unknown sequences, where the user can get a good idea of how trustworthy the predictions are. Finally, WAR makes it easy to download the alignments and structures, both for the individual programs and the consensus, for further analysis.

It should be stressed again that the performance of the individual methods depend strongly on the actual dataset: the RNA family, the sequence length, the overall identity—all of this will affect the performance. The strength of the WAR server is the ease with which the different methods can be compared on different RNA datasets. In the example above, the tRNA dataset is fairly divergent and will thus be hard for methods that rely on a good, purely sequence-based alignment. A set of more closely related sequences will on the other hand be a challenge to programs that need a strong evolutionary signal from covariation. The goal of the above example is not to compare the individual methods but to show how the server works. On the website, the result of a different, less divergent dataset is available as an example.

**ACKNOWLEDGEMENTS**

We thank Jan Gorodkin, Stefan Seemann and Jakob S. Pedersen for their comments, and Paul Gardner for some rewarding conversations. Funding to pay the Open Access publication charges for this article was provided by a faculty PhD stipend.

**Conflict of interest statement.** None declared.

**REFERENCES**


Introduction

This second part of my dissertation will introduce an area distinct from the RNA part. Next generation sequencing (NGS) is a field in rapid development, and various different platforms exist with their individual strengths and weaknesses. I will mainly focus on the Illumina sequencing platform. This part will introduce the field of mapping short reads to a reference genome and performing genotyping and SNP calls from the assembly. A first author paper (draft version) presents a method for genotyping and SNP calling using a generative probabilistic graphical model (chapter 4). The main result from my work in this field led to a joint first author paper presenting the first sequence of an ancient human genome based on DNA extracted from an approximately 4000 years old individual from Greenland (chapter 5). This project will be referred to as the Saqqaq project.

Next generation sequencing

In 1990, the Human Genome Project was initiated. Before this time only small genomes like that of Bacteriophage MS2 [FCD+76] were known with a size that is merely a small fraction of the human genome. The goal was to obtain the DNA sequence of the human genome and begin work on understanding the complexity of genomics. It was a daunting task. The complete sequence was released more than a decade later [JR03], and the cost of the project had to be measured in billions of dollars.

A few years later, the complete diploid genome of a single human individual was published [LSN+07]. Today, there are more than 100 eukaryotic genomes in the databases – and even more prokaryotic genomes – and the so-called 1000 genomes project is well under way at a cost that is a small fraction of the original Human Genome Project. The technological advances and resulting lower prices within the field of DNA sequencing has led to an explosive growth in the number of genomic sequences available. This huge amount of data is arguably one of the most important developments in the last decade, giving rise to a number of exciting projects and deepening our understanding of the complexities of life. Recent advances have led to the first sequences of now extinct
species being obtained such as the woolly mammoth [MDR+08] and the first analysis of genomic data from neandertal [GKP+06, NCK+06].

One of the major developments in sequencing technologies is the appearance of so-called next generation sequencing (NGS) techniques. This covers a spectrum of technologies that all differ from Sanger sequencing [SC75]. In Sanger sequencing, DNA sequences are obtained by using a single stranded DNA template and a mixture of DNA polymerases, standard deoxynucleotides and labelled dideoxynucleotides. The DNA template is sequenced, until one of the dideoxynucleotides is incorporated into the new DNA-sequence. These nucleotides lack the 3’ OH-group and thus cannot form the phosphodiester bond needed to add the next nucleotide to the sequence. The result is a mixture of DNA sequences of varying lengths which can be ordered by length, and the labelled dideoxynucleotides can then be used to determine the DNA sequence. This process can generate DNA fragments of appr. 500-700 nucleotides.

NGS techniques work on a different principle where the DNA polymerase reaction is observed one step at a time, and many DNA templates are sequenced in parallel [SJ08, Met10]. In general, the single stranded DNA templates are immobilized, and the new sequences are extended stepwise. In e.g. 454 pyrosequencing [MEA+05], a solution containing DNA polymerase and luciferase is used, and at each step only one of the four nucleotides is added. Wherever a nucleotide is incorporated into the new sequence by the polymerase, a subsequent reaction with the luciferase will produce a burst of light. If a sequence has a stretch of identical nucleotides, they will be incorporated in the same cycle, and the burst of light will be proportional to the number of incorporated nucleotides. Thus, in each step the DNA templates that matched the nucleotide added will light up, and the sequences are recorded stepwise by a camera.

A different approach is taken by Illumina [FRW+06, TRFT08] where a mixture of all four nucleotides with different labels is added at each step, and the reactions are recorded in four different channels. The Illumina platform works by immobilizing the DNA templates on a glass slide in so-called clusters. Each cluster consists of around 1000 copies of the same DNA template, and there are millions of clusters – each corresponding to a different DNA template. The clusters are built from single templates that are subsequently amplified on the slide using bridge PCR. Thus, massive parallel sequencing is possible, generating millions or billions of nucleotides. The initial DNA templates come from a PCR library that has been created based on the sample in question. The PCR products are added to the flowcell, and further amplification takes place where the sequences are being tethered to the surface. An Illumina flowcell has of 8 lanes (with one lane being reserved for control runs), and each lane as mentioned contains millions of clusters.

The nucleotides used in the Illumina approach have been modified to contain one of four fluorescent labels (to distinguish between them in a single cycle) and a removable terminator in the 3’ end that stops the polymerase from incorporating another nucleotide. In that way, only a single nucleotide is incorporated in each step, and the label makes
it possible to see which one (compared to e.g. 454, where multiple nucleotides can be incorporated in a single step, giving rise to problems with nucleotide homopolymers). By measuring the light in the four channels, the Illumina machine assigns a probability to each of the four nucleotides, and the most likely one is called as the correct one. After each cycle, both the fluorescent label and the terminator group are cleaved, and another sequencing step is performed. At present, sequences of appr. 75 nucleotides can be generated in this fashion without too much loss in quality, compared to appr. 35 nucleotides just one year ago.

NGS generates a huge number of nucleotide sequences, but errors occur. For each step, the signal-to-noise ratio decreases. Also, any error that happened at an earlier step will degenerate the signal more and more as the process continues. The nucleotide sequences generated are therefore often matched with a sequence of qualities that can be seen as the probability of an error being made at each step – i.e. we do not just get a nucleotide \( N \), we are also given a quality \( Q \) corresponding to the probability that the actual nucleotide in the sequence is not \( N \).

For Illumina data, the Phred [EG98] quality scores are given as a sequence of ASCII characters, and the numerical value can be translated into an error probability. The ASCII string is a convenient representation of the quality scores where some offset \( \Delta \) is used to get a span of printable characters. For an ASCII character \( Q \), the actual quality score is therefore \( SC(Q) = \text{ord}(Q) - \Delta \). Normally, either \( \Delta = 33 \) or \( \Delta = 64 \) is used, and the quality scores go from 1 to 40. The error probability is given as:

\[
P_{\text{error}}(Q) = 10^{-\frac{SC(Q)}{10}}
\]

The higher the quality score \( SC(Q) \), the lower the probability of an error. This quality information is important for performing correct mapping of the reads and is also useful for calling the correct genotype.

Using one of the NGS techniques, it is possible to generate either single end reads (i.e. all the DNA templates are sequenced from one end) or paired end reads (where a template is sequenced in both directions in two separate runs). Single end reads are cheaper to generate, but paired end can give more information about the DNA being sequenced – either by generating sequences from both ends of a long fragment making the subsequent mapping or assembly easier, or by sequencing the same short sequence twice in both directions hence improving the confidence in the consensus. Furthermore, since there is a decrease in quality for each sequencing step, paired end reads also have the advantage for short fragments of sequencing the templates in both directions, thus providing high-quality sequences of both the 3’ and 5’ ends of the templates. Single end reads are used in the Saqqaq project.

Obtaining the reads using one of the NGS techniques is only the first step. The read sequences will have to be quality checked in different ways, then they have to be assembled or – if a reference genome is known – mapped to the reference, and then genotyping can be performed where the original DNA sequence is inferred. Each of
these steps is challenging and will be elaborated below. In general, the focus will be on mapping reads to a human reference as this is the goal in the Saqqaq project.

**Ancient DNA**

The possibility of working with ancient DNA in the lab is relatively new, dating back only a few decades [HBF+84, Pää85, PHW89]. Within recent years many significant improvements have been made, making it possible to work not only with short mitochondrial DNA sequences but also with nuclear DNA from the specimen. This opens an interesting field of research that makes it possible to directly investigate the genetics of the past, thus being of growing importance in the fields of population genetics and evolutionary biology [WC05].

Dealing with ancient DNA presents some unique challenges that need to be overcome in any ancient genomics project. The collection of the sample and extraction of the DNA is challenging, and it is of the utmost importance that these steps be carried out correctly in order for all of the subsequent analyses to be of any value. Here, the focus will be on the handling of the data and not the preparation of the sample. Some of the key issues when dealing with ancient DNA are damage, fragmentation and contamination. On top of this we have all the normal sources of error that affect any genomics project such as incorrect base calls. An interesting recent development is the use of pathological samples that have been formalin-fixed and paraffin-embedded. This treatment damages the DNA, but recently samples have been used in sequencing projects [SKT+09] opening up for new medical applications.

Since ancient DNA has obviously not been stored efficiently with the goal of eventually being sequenced, it has normally been exposed to a number of degrading processes. These post mortem sources of damage make the isolation and sequencing difficult and also affect the base calling. Some sources of damage lead to cross-linking in the DNA which interferes with the polymerase in the PCR. The most abundant source of DNA damage, though, is deamination of cytosines. Through hydrolysis, ammonia is released and the cytosine nucleotide is turned into uracil – a nucleotide present in RNA but not part of DNA. In living cells, uracils in the DNA would be recognized as errors and removed, but this correction does of course not happen post mortem.

Depending on the polymerase used when analyzing ancient DNA, uracils will show up as the nucleotide thymine in the PCR. The cytosine-to-uracil damage will therefore result in a higher number of $C \rightarrow T$ or $G \rightarrow A$ substitutions (depending on the strand) than expected by chance. This will affect the genotyping and lead to wrongly called single nucleotide polymorphisms (SNPs) if not dealt with properly. Overlapping fragments being sequenced will not necessarily all be damaged to the same degree. Thus, one way to locate possible damaged sites and estimate the damage rate is to look for an overrepresentation of $CT$ and $AG$ SNPs since they could be due to damage rather than real mutations.
Over time, long DNA strands will fragment. This gives rise to issues with sequence lengths in the analysis. Using Illumina, a huge number of sequences will be generated in parallel all of length $L$. If the actual genomic DNA being sequenced is shorter than $L$ nucleotides, part of the reported sequence will actually be the adapter sequence used in the reaction and not the genomic DNA. However, it cannot be seen from raw reads whether the nucleotides are genomic (ancient) DNA or part of the (new) adapter sequence.

If this problem with adapter inclusion is not fixed, the mapping can result in a large number of wrong SNP calls as a large fraction of non-human DNA will be forced to map to the reference. Assume that a read of length $L$ contains $L_G$ nucleotides from genomic DNA and $L_A = L - L_G$ nucleotides from the adapter. The $L_G$ nucleotides can be mapped correctly to the reference, forcing the $L_A$ adapter nucleotides to be mapped next to the genomic DNA. Since these nucleotides do not match the human genome except by chance, it will give rise to potentially $L_A$ wrong SNP calls.

Obviously, this has to be dealt with. Due to the sequencing reaction, any included adapter sequence will always be in the (low quality) 3’ end of the read. Since the adapter sequences are known, it is possible to search for adapter inclusion in the reads. This can be done using pairwise local alignment as described in the first part of this dissertation. However, a certain number of nucleotides is needed before sequence alignment can discover the adapter fragment. A rule of thumb says at least 6 nucleotides.

In case of single end reads, this process can be done by aligning the 3’ end of the read to the 5’ end of the reverse complement of the adapter sequence. The longest match can be found and removed. Of course, one should take into account that errors are made in the base calling of the reads – especially in the 3’ end – so strict identity will be too harsh a criterion. It might be better to make a more relaxed alignment, allowing for mismatches, in order to remove as much adapter as possible at the cost of also removing some real genomic DNA. This is a trade-off between including adapters and making wrong SNP calls, and throwing away real data. To produce useful results, it is often better to be strict and throw out a little too much than including wrong DNA. Hence, matches of lengths down to 1 could be removed to be on the safe side, although many
of these will be purely random. It would also be possible to use the quality scores in the alignment procedure, allowing for more mismatches if the quality scores are low. A different approach that will be discussed in the next section is to make the adapter removal part of the mapping procedure.

If paired end reads are used, the extra information can be used to make the discovery and removal of adapter sequences more efficient. When working with ancient DNA, the fragments are often only 30-70 nucleotides long. Thus, when a fragment is sequenced using paired ends, both reads will often contain the full genomic sequence and part of an adapter. In that case, we have two reads of length $L$, both covering the same genomic DNA of length $L_G$, and both having $L_A$ nucleotides of adapter sequence in the 3’ end. In the paired end setup, two different primers are used, so it will be two different sequences of length $L_A$. One way to solve this problem is described below with references to Fig. 6.

![Diagram](image)

**Figure 6**: The procedure to remove included primer fragments from next generation paired end reads. See main text for details.

Fig. 6A shows the original construct: The genomic DNA of interest with two flanking adapters, designated PCR1 and PCR2. In case the region denoted GENOMIC is shorter than $L$ nucleotides, both of the two paired reads will contain part of an adapter. In Fig. 6B, this is illustrated: READ1 contains a genomic part and part of the PCR2 adapter, which will be the reverse complement of the actual sequence (hence the name RC-PCR2). Similarly for READ2. The two genomic sequences will be the reverse complement of each other. The problem of adapter inclusion can therefore be solved by
aligning e.g. READ1 to the reverse complement of READ2. If the genomic part overlaps between the two, the remaining overhang will be adapter sequence.

This observation is used as illustrated in Fig. 6C: If we have inclusion of adapter sequence, the 5’ end of the reverse complemented READ2 will match with the 3’ end of PCR1. Likewise, the 3’ end of READ1 will match with the 5’ end of the reverse complement of PCR2. If we concatenate PCR1 with READ1, and RC-READ2 with RC-PCR2, we can perform local pairwise alignment between the extended sequences and thus find not just the genomic overlap but also the included adapter sequences as illustrated in Fig. 6D. The maximal alignment found in the process can be subdivided as shown in Fig. 6E, and the genomic sequences are extracted and used in the subsequent mapping. This approach makes use of all available information and can discover even the inclusion of a single nucleotide from the adapters with high confidence.

A third problem encountered with ancient DNA is contamination. It is extremely difficult to be sure that what you sequence is actually the ancient DNA and not the people who have been handling the specimen. In fact, a large part will most likely be microbes and other sources of DNA from the environment. There are ways to minimize contamination or make it easier to spot in the analysis. Of course, isolation and proper handling is key. In the present study, we used DNA extracted from hair [GTR+07] where the DNA has been more protected from the environment and exogenous DNA contamination is minimized. Once in the lab, the DNA fragments had a barcode of 6 nucleotides introduced in the library preparation. In all downstream analyses only sequences with the correct barcodes were used to eliminate any contamination following the DNA extraction and purification.

Up until the laboratory work, the sample had been handled primarily by Europeans, making it possible to look for European contamination in what is expected to be a genome more similar to Asians. In short, this was done by collecting a set of alleles private to Europeans (i.e. not known to appear in Asian individuals). An analysis was done on the positions in this set covered in the sequenced genome. By observing the number of times the ancient genome matches the European allele compared to the estimated error rate the level of contamination can be estimated.

**Mapping reads to a reference genome**

After the sequencing step and post processing, dependent on whether the genome is known or not the next step is either to map the sequences to the genome or perform *de novo* assembly. The latter is not done for ancient DNA. The reads can be of varying lengths and from various places in the genome. Some parts of the genome can not be resolved due to repetitive sequences, while others will be sequenced multiple times by many different overlapping fragments. *De novo* assembly is the task of solving this giant jigsaw puzzle. How should all these small DNA pieces be put together in order to reconstruct the entire genome? This is done by finding identical subsequences where the
fragments overlap by aligning the sequences, combining the two overlapping fragments, and thus constructing longer and longer sequences. Since sequencing errors occur, part of the process is to deal with variations in the sequences that can not be explained by different alleles but must be due to errors – this is part of the genotyping step which will be explained in the next section.

If a reference sequence exists from a previous sequencing project, we have to solve the much easier mapping problem. Again, we have a large number of reads of varying lengths and from various positions in the genome. The goal is now to align the reads to the reference and find the best match. Based on the reference, all the reads are aligned creating a map of the genome where some positions have many overlapping reads aligned to it while others are more shallow or even lacking reads completely. Whether mapping or assembly is being performed, some common terms are used to describe the result. The coverage is the percent of the total genome sequence that is covered by the reads, and the depth is the number of reads and hence nucleotides covering a specific position.

Compared to Sanger sequencing, NGS generates short reads which complicates the mapping. The difficulty of mapping a read is directly related to the information content of the sequence relative to the genome, which is mainly affected by the length of the read, and how repetitive the sequence is. The shorter a read is, the larger the potential number of hits. To resolve ambiguities unique hits are preferred, but the existence of repeat regions combined with short read lengths lead to some reads mapping to multiple locations. How to deal with multiple hits is important since it can skew the coverage. With a huge number of reads such as that generated using Illumina, it becomes intractable to perform full pairwise alignment between the reads and the genome. Instead different optimization shortcuts are taken. A widespread way of simplifying the mapping is to focus on the high quality 5' end of the reads and only allow a maximum number of mismatches.

In MAQ [LRD08], the reads are indexed using a hash table, the genome is scanned for matches, and the matches are extended and scored. The first and most accurate 28 nucleotides of the reads are indexed and scanned against the genome, thus guaranteeing alignments with up to 2 mismatches in the first 28 positions. The process of hashing the reads into a table of integers speeds up the scanning compared to doing actual pairwise alignment. When a hit is found in the genome, the mapping is extended to the full read length beyond the 28 nucleotides without introducing gaps. The score \( q \) is calculated based on the sum of the quality scores of the mismatched positions over the full read.

MAQ generates six hash tables, each containing hash values for all reads, using six templates for the seed region that are designed to find all alignments with up to two mismatches in the seed. Allowing for more mismatches can be achieved at the cost of speed by introducing more templates. The genome is scanned with the templates in pairs on both forward and reverse strands. If a read maps equally well to multiple positions, a random hit is chosen as the result.
For each mapped read, MAQ assigns a mapping quality \( Q_m \) similar to Phred scores as described previously. These scores are used to calculate the posterior probability that the read is wrongly mapped (i.e. high scores give low probabilities). The error probability is proportional to the qualities of mismatches, and thus by minimizing the summed mismatch qualities MAQ effectively maximizes the posterior probability of correct mapping.

Another widely used mapping tool is Bowtie [LTPS09]. This is a very fast program with a reasonable memory usage. Here, the reference genome is indexed and the generated index file can be reused. Similar to MAQ there is a limit on the number of mismatches allowed in the high quality 5’ end of the read, and Bowtie also limits the total number of mismatches by having a maximum allowed value of the sum of mismatch qualities.

Bowtie uses the Burrows-Wheeler (BW) transformation [BW94] of the genome sequence to build the index, which can then be searched efficiently with a fairly low memory footprint. Shortly, the BW index is created by appending a character $ to the search text that is not part of the alphabet and is defined to be lexicographically smallest. The text is cycled to create all possible splits of the original text, using $ as the border character. The sequences are then sorted lexicographically, and the BW transformation of the search text is comprised of the rightmost column of the matrix thus created. The BW matrix has a characteristic known as “last first mapping” which can be used when matching a query sequence against the search text to find increasingly longer suffixes of the query.

Standard search in a BW matrix finds exact matches. To allow for mismatches between the read and the genome, Bowtie has altered the search routine to find inexact matching suffixes. When the exact suffix match cannot be increased, Bowtie chooses an already matched suffix, introduces a mismatch on the next position, and continues the exact mapping after the mismatch. This is done in a way that minimizes the sum of quality scores at the mismatched positions. The default search strategy in Bowtie will not necessarily return the best alignment, but options exist that make it possible to search deeper at the cost of time.

For the Saqqaq project, we developed a novel mapping tool named Sesam (Search Enhanced Suffix Array for Mappings) (Krogh 2009, unpublished). As the name implies, it uses an enhanced suffix array (ESA) to speed up the procedure [BHGK06]. An ESA is a datastructure based on suffix trees, but while suffix trees require a lot of memory, ESAs are much smaller since they do not store the tree structure itself. Instead, an ESA is a list of the suffixes in the search text ordered as they appear as leaves in the suffix tree when doing a depth first search. The ESA also contains information on how suffixes are related in the tree.

It takes time to build an ESA, although the asymptotic build time is \( O(L) \) in the sequence length \( L \), but if the index is going to be searched multiple times – as is the case with sequencing projects – the build time can easily be made up for by the actual
searches. The ESA is constructed by lexicographically ordering all suffixes of the text, which is equivalent to doing a depth first search of the suffix tree. The ESA also contains information on the longest overlap between a prefix of the current entry and a suffix of the previous entry (or longest common prefix, \(lcp\)). A final piece of information in the ESA is the so-called skip column that points to the next entry to visit in a depth first search that has a smaller \(lcp\).

In \textit{Sesam}, the reads are represented as position specific scoring matrices (PSSM). At a given position, the probability of the read nucleotide \(N\) is \((1 - P_{\text{error}}(Q))\) where \(Q\) is the quality score. The PSSM entry for this nucleotide is \(\log_2 \left( \frac{1-P_{\text{error}}(Q)}{q(N)} \right)\), and the PSSM entries for the remaining three nucleotides are \(\log_2 \left( \frac{P_{\text{error}}(Q)}{3q(N)} \right)\), where \(q(N)\) is the background frequency of nucleotide \(N\) in the genome. In the search through the ESA, only reads scoring at least 50\% of the maximum possible score are reported, and to avoid spurious hits a stringent mapping of the first 20 nucleotides is performed.

For each read, a specific match \(i\) has a probability of being correct which is estimated as \(P_i = \frac{s(i)}{\sum_j 2^{s(j)}}\), where \(s(j)\) is the score of the \(j\)'th match of the read. This estimates the probability of the \(i\)'th hit being correct by comparing the score \(s(i)\) to the sum of the scores of all hits \(j\). In principle, the sum should be over all possible hits of the read which is of course intractable. Only matches with a probability of at least 90\% are reported. To address the problem of repeats, a read with more than 100 matches above the threshold is also ignored.

As part of the mapping procedure, \textit{Sesam} also finds and removes included adapter fragments. This is done by mapping the read to the reference genome up until a position \(i\) in the read. By checking if the read aligns better to the adapter sequence than the genome after position \(i\), included adapter fragments can be found and removed.

Using an ESA, the theoretical maximum coverage one can obtain on the human genome was calculated. Due to repeats some regions of the genome will be impossible to map uniquely, and this fraction will become larger for shorter reads. For a given read length \(L\), a repeat of length \(2L - 2\) can be recovered in the perfect case if two reads map to separate halves with a one nucleotide overhang to anchor them in the genome. Using this approach, a conservative estimate of the repetitive fraction of the human region is calculated to be 15.2\% for length 50 and 13.3\% for length 60.

### Genotyping

When all the reads have been mapped to the genome, the next step is to determine the genotype – i.e. the diploid genome sequence in case of human DNA – of the individual being sequenced. The reads will not be evenly distributed over the entire genome, so some regions will have high read depths, other regions low, and some will not be covered at all. For each genomic position covered, the goal is to infer the original genotype in
the individual based on the reads.

Since the human genome is diploid (except for the X and Y chromosomes (for males) and the mitochondrial DNA) not all positions will necessarily be covered by the same nucleotide. Some positions in the genome are homozygotes, meaning that the two alleles are identical, and others are heterozygotes, i.e. with different alleles. Some positions might be covered by a high number of identical nucleotides, making it easy to infer the original homozygote genotype. Others might have close to a 50/50 split between the two alleles. However, this is often not the case. Rather you will have some number of all nucleotides due to uneven sampling, wrongly mapped reads and errors in the base calling step. In the case of ancient DNA, there is also the issue of DNA damage that should be dealt with.

The goal in genotyping is to get around these sources of errors and determine what the original data looked like by looking at the read data we have. The quality scores can be used directly to assess how much confidence to have in each of the called nucleotides. One could also use the information from a reference sequence to assist the genotyping process since this extra prior knowledge can help make the decision about the final genotype – especially with very noisy data or low read depths. One goal of this process is of course to detect single nucleotide polymorphisms (SNPs).

In large scale genomics studies, one of the goals is to estimate the population substructure – i.e. groups within the larger population between which the gene flow is limited compared to the gene flow within a group. This can be estimated from SNP data, since a group with a large SNP overlap indicates a subpopulation compared to other groups. Such analyses have been carried out on European populations [SSV06] and on the large population of Han Chinese [XYL09]. This sort of information can also be used to place an individual in a population based on the observed SNPs as in the contamination analysis mentioned previously.

A widely used method for genotyping and SNP calling is part of the MAQ program [LRD08] – now included in the SAMtools package [LHW09] – mentioned in the previous section. MAQ assumes a diploid genome and calls the genotype at a given position based on the two most frequent nucleotides, thus only considering three combinations at each site (i.e. for the two most frequent nucleotides a and b, the possible genotypes are aa, bb and ab). Assuming a prior probability of a SNP to be $r = 0.001$ by default, the prior for one of the two homozygote genotypes becomes $P(aa) = P(bb) = \frac{1-r^2}{2}$.

As part of the mapping process in MAQ, each read received a mapping quality $Q_m$ similar to the base call qualities assigned to each nucleotide. Since this score corresponds to the probability of a wrongly mapped read, this information is included in the genotyping. If the read is mapped incorrectly, you cannot infer genotypes from it, and thus all qualities higher than $Q_m$ is set to be equal to $Q_m$, i.e. the lower quality is always used. The posterior probability of each genotype is calculated, and the one with the highest posterior probability is reported.

To address the specific problems encountered in the data we developed our own
genotyping and SNP calling program called SNPest (Single Nucleotide Polymorphism estimation). The aim was to have a sensitive genotyping tool that could avoid systematic biases due to e.g. read errors without being specifically tailored for ancient DNA. SNPest is a probabilistic model that takes both quality scores and alternative sources of read errors explicitly into account. The method is based on a generative model of the probability distribution over genotypes given the sampling and sequencing of nucleotides obtained from a diploid genome. In chapter 4 a draft version of a paper describing SNPest is presented.

The graphical model used in SNPest is illustrated in Fig. 7, where circles represent random variables. This is a generative model, describing the sampling process from the top down. $H$ is the reference nucleotide in the human genome (normally a flat prior is defined on $H$, which may be observed). $G$ is the genotype in the individual at the given position that we want to infer. The genotype distribution depends on the reference nucleotide. The random variables in the box model the $n$ sampled reads covering the given position. Each sampled read contains an original nucleotide dependent on the genotype ($S_i$). The original nucleotide is called as a possibly different nucleotide in the read ($R_i$). The read nucleotide gives rise to the actual light intensities observed in the sequencing machine ($I_i$).

![Diagram of the graphical model](image)

**Figure 7**: The graphical model that forms the backbone of SNPest. The two top random variables are global for a given position in the genome, while the variables in the box model the $n$ reads covering the position. The random variables are the reference nucleotide in the human genome ($H$), the genotype ($G$), the original sampled nucleotide ($S_i$), the read nucleotide ($R_i$), and the light intensities ($I_i$).

This models a probability distribution over a set of random variables as denoted by the circles in the graphical model. The observed random variables are the light intensities ($I_i$) and the reference nucleotide in the human genome ($H$). From these, the unobserved
random variables are inferred: The nucleotide in the read sequence ($R_i$), the sampled nucleotide ($S_i$), and the genotype in the diploid genome ($G$). For a single position in the genome, the combined probability distribution can be written as:

$$P(H, G, A, R, I) = P(H)P(G|H)\prod_{i=1}^{n} P(S_i|G)P(R_i|S_i)P(I_i|R_i)$$

The product is over the $n$ reads covering a specific position, with variables $S_i$, $R_i$ and $I_i$ corresponding to each of the reads. The probability factorizes into a five separate probabilities, each of which can be interpreted in terms of sampling nucleotides from a given position in the ancient diploid genome.

$P(H)$ is the uniform prior distribution over the reference nucleotide in the human genome. $P(G|H)$ is the conditional probability distribution over the ten possible genotypes \{AA, AC, AG, AT, CC, CG, CT, GG, GT, TT\} given the nucleotide in the reference genome. $P(S_i|G)$ is the conditional probability distribution over the sampled nucleotides given the genotype at that position. $P(R_i|S_i)$ is the conditional probability distribution over the read nucleotides given the original sampled nucleotide. $P(I_i|R_i)$ is the conditional probability distribution over light intensities observed in the sequencing machine given the read nucleotide.

The conditional probability distribution $P(G|H)$ was estimated from the diploid Yanhuang genome [WWL+08] where genotypes were counted for each reference nucleotide. $P(S_i|G)$ is the probability of having a specific nucleotide in the original sampled DNA fragment given the genotype. If the genotype is a homozygote in nucleotide $N$, the probability for $N$ is 100% and 0% for the three other nucleotides. If it is a heterozygote, the probability is 50% for the two alleles and 0% for the remaining two nucleotides.

Due to damage and various sources of error, the actual nucleotide in the individual might not have been what we observe in the read. Since the experimental setup in the Saqqaq project uses Phusion polymerase, uracils block transcription and hence most damaged positions are not included in the reads [FPC02]. The probability $P(R_i|S_i)$ models all other sources of error e.g. from the PCR amplification, adapter sequences not removed previously, wrong mappings etc. The error rate was estimated to be 0.33% by counting the observed number of mismatches from read nucleotides with high quality. We do not know the form of the conditional probability distribution $P(I_i|R_i)$. However, this probability is proportional to $P(R_i|I_i)$ which is given by the quality scores.

$SNPest$ estimates the genotype at a given position to be the one with the highest posterior probability. This probability is found by marginalizing the combined probability distribution and summing out the $S$ and $R$ parameters. This gives us the probability $P(G, I, H)$, and by marginalizing $G$ we can calculate $P(G|I, H)$.

A haploid version of $SNPest$ was specified to deal with chromosomes X and Y and the mitochondrial DNA. This only required a few changes. The probability distribution over the genotype now only includes the four haploid genotypes \{A, C, G, T\}, the
conditional probability distribution over genotype given the reference, \( P(G|H) \), is estimated from the Yanhuang X chromosome only, and the probability distribution over the original nucleotide given the genotype, \( P(S_i|G) \), is simplified to be the identity matrix.

The input to \( I \) in \textit{SNPest} is calculated from the raw reads. For each position in the sequenced genome, there is a number of nucleotide and quality score pairs. Each pair corresponds to a probability distribution over the four nucleotides. Given a quality \( Q \) and a nucleotide \( N \), the probability that the read nucleotide is actually \( N \) is \( 1 - P_{\text{error}}(Q) \), and the probability that it is either one of the three other nucleotides is \( P_{\text{error}}(Q) / 3 \) for each.

This distribution can be precalculated as a \( 4 \times |Q| \cdot 4 \) matrix, where \( |Q| \) is the number of possible quality scores. A given \((N, Q)\) pair corresponds to picking a particular column in the matrix. The reference nucleotide \( H \) is a flat prior with probability \( 25\% \) for each nucleotide that can be observed as well.

In the model, there are two parameters that can be tuned to fit the data. First, the prior on the genotype is estimated from a genome with an average SNP frequency of 0.12%. However, if the dataset is known to have either a higher or lower frequency of SNPs, it would be appropriate to tune this prior to fit the data. This should be done such that for each nucleotide \( N \), only the overall SNP frequency is changed and not the relative frequencies of the 9 other genotypes.

Let \( \mathcal{F} \) be the SNP-frequency:

\[
\mathcal{F} = 1 - \sum_{N \in \{A, C, G, T\}} P(N)P(N, N)
\]

where \( P(N) \) is the prior for the four nucleotides – by default, a flat prior \( P(N) = 0.25 \) is used – and \( P(N, N) \) is the probability of seeing the homozygote genotype \( NN \) as estimated from the data. We want to introduce a tuning parameter \( \alpha \), such that we can define the SNP frequency \( \mathcal{F} \) we expect to see in the data, i.e.:

\[
\mathcal{F} = 1 - \alpha \sum_{N \in \{A, C, G, T\}} P(N)P(N, N)
\]

The parameter can be found as \( \alpha = \frac{1 - \mathcal{F}}{\sum_{N \in \{A, C, G, T\}} P(N)P(N, N)} \). This parameter scales the probability of seeing genotype \( NN \) for each nucleotide \( N \). To scale the remaining 9 probabilities for each nucleotide, the remaining probability mass after scaling \( P(N, N) \) with \( \alpha \) has to be redistributed such that all 10 probabilities sum to 1:

\[
1 - \alpha P(N, N) = \beta_N \sum_{N' \neq N} P(N, N')
\]

where the sum is over \( N' \in \{A, C, G, T\} \) different from the current nucleotide \( N \). The nucleotide specific tuning parameter \( \beta_N \) is then given as:

\[
\beta_N = \frac{1 - \alpha P(N, N)}{\sum_{N' \neq N} P(N, N')} = \frac{1 - \alpha P(N, N)}{1 - P(N, N)}
\]
for each of the four nucleotides $N$. Thus, specifying a SNP frequency $\mathcal{F}$ directly yields a tuning parameter $\alpha$ for the four homozygote genotypes equal to the reference nucleotide and four $\beta$-parameters for scaling each row in the matrix. Another option is to use a flat prior on the genotypes, where the probability of observing any of the 10 genotypes is set to 10%.

The input to the model treats the quality measures as being correct – i.e. the nucleotide probability distribution is calculated directly from the quality score. However, read errors, wrong mappings etc. will add noise to the data as given by $P(R_i|S_i)$, where the distribution includes different sources of error as described above. If we expect an error rate of $\tau$, we multiply the length 4 vector $\vec{P}_R$ describing the nucleotide distribution $P(R_i|I_i)$ by an error matrix $\mathbf{A}$, yielding the new probability vector $\vec{P}_S$:

$$\vec{P}_S = \mathbf{A} \ast \vec{P}_R$$

where

$$\mathbf{A} = \begin{bmatrix}
(1 - \tau) & \frac{\tau}{3} & \frac{\tau}{3} & \frac{\tau}{3} \\
\frac{\tau}{3} & (1 - \tau) & \frac{\tau}{3} & \frac{\tau}{3} \\
\frac{\tau}{3} & \frac{\tau}{3} & (1 - \tau) & \frac{\tau}{3} \\
\frac{\tau}{3} & \frac{\tau}{3} & \frac{\tau}{3} & (1 - \tau)
\end{bmatrix}$$

By default, the error rate is estimated to be $\tau = 0.33\%$ but this tuning parameter can also be set to reflect the expected error rate in the data.

This error correction uniformly distributes the errors over all probabilities. In reality, some read errors occur more often than others. This can be modelled by estimating the actual errors. This was done by collecting a set of high quality nucleotides from the large Saqqaq dataset where we expect an error rate of $P_{error}(Q) = 0.04\%$ or lower. For each position, the correct nucleotide was decided by majority vote, and the number of discrepancies was counted as the fraction of nucleotides that differ from the true answer. To avoid counting actual heterozygote positions as errors, this was done on the two sex chromosomes. The estimated error matrix looks like:

$$\mathbf{A}_{Saqqaq} = \begin{bmatrix}
0.9982 & 0.0003 & 0.0010 & 0.0004 \\
0.0006 & 0.9979 & 0.0004 & 0.0011 \\
0.0010 & 0.0003 & 0.9980 & 0.0006 \\
0.0004 & 0.0010 & 0.0003 & 0.9983
\end{bmatrix}$$

SNPest is able to take different sources of information into account. This includes errors as discussed previously but also extra confidence in specific sequences can be used. During PCR amplification, the problem of clonal expansion can occur. This happens when one or a few DNA sequences happen to get ahead in the PCR reaction. After a few steps in the exponential process of PCR, they will dominate the mixture. When sequencing the DNA strands, these clonal sequences will be reported as separate reads although they are actually identical sequences from the same template. If this is not dealt with they can significantly bias the genotyping process.

Clonal reads can be identified after mapping by looking for sequences that 1) have the same length, 2) map to the same genomic position, 3) map to the same strand, and
4) are from the same PCR reaction. If all four demands are met, the sequences can be grouped together as a “clonal family”. There are different ways of dealing with this. You could keep one representative sequence (e.g. the best in terms of quality scores) and ignore the rest. A better solution that we pursue is to use all the information present in the large collection of clones and instead calculate a representative read sequence.

This collapsed clone representative sequence is calculated for each clone family by creating a $4 \times L$ matrix, where $L$ is the read length of the clones. For each position $i = 1, 2, \ldots, L$, the sum of the log-transformed probabilities of observing each of the four nucleotides in each of the clones is calculated. Based on this matrix, the most likely sequence representing all the clones is found by choosing the most likely nucleotide at each position. The corresponding qualities are calculated from the summed log-transformed probabilities.

These new error probabilities can be much lower than the $10^{-4}$ that is the minimum in the Illumina pipeline. We therefore extend the quality range by an additional 10 characters, allowing error probabilities down to $10^{-5}$. In principle, the range could be extended even further. However, possible errors from e.g. the PCR reaction would get a lot of weight in the SNP calling if they occurred in a clone family, so to avoid this bias the scale is truncated. These new qualities can be treated directly in SNPest simply by extending the input matrix representing $P(R_i|I_i)$.

In the paper in chapter 4, SNPest is tested with different tuning parameters and compared to the performance of SAMtools, showing that SNPest is comparable. SNPest was also used in the Saqqaq project as described in the following section.

**First diploid genome of an ancient human individual**

A major project involving next generation sequencing of ancient DNA followed by bioinformatical analyses of the reads was done on a hair sample obtained from the permafrost. This belonged to an individual from the Saqqaq culture, the first people to settle in the New World Arctic. The work resulted in the first ancient genome obtained from a human individual.

Due to the age of the sample of approximately 4000 years, we expected to encounter all the problems described above when working with ancient DNA. However, the use of the Phusion polymerase excluded damage due to deamination of cytosine [FPC02]. Our group was in charge of handling the data and performing all steps of quality control and bioinformatical analyses including genotyping and SNP calls.

The mapping of the reads was done using the Sesam program outlined above, and genotyping was done with SNPest. Problems with inclusion of adapter sequences and clonal expansion in the PCR amplification were solved as explained in the previous sections.

Genotyping/SNP calling was done on the entire genomic sequence. A high confidence subset was generated from this containing positions where the prediction is par-
particularly good. The high confidence set is split into positions similar to the reference genome, and positions differing from the reference, i.e. the SNP set.

The high confidence set similar to the reference was created by demanding 1) the posterior probability of the genotype is at least 99.99%, 2) the position is not in an annotated repeat, and 3) the read depth is between 10X and 50X. For the SNP set, we further demand that the distance to the nearest neighbouring SNP is at least 5 nucleotides. The restriction on read depth was enforced to avoid positions lacking data (i.e. less than 10 reads) and to avoid problematic areas due to un-annotated repeats (i.e. more than 50 reads). The neighbour restriction on SNPs was enforced to avoid calling wrong SNPs due to insertions and deletions. The mapping cannot locate indels, so just a single nucleotide indel can create a cluster of erroneously called SNPs on positions following the indel – simply because the remainder of the read is forced to align in an offset fashion.

The data presented in chapter 5 is of good quality considering the age. We recover 79% of the genome with an average read depth of 20X. Since the average read length is 55 nucleotides after removing included adapter sequences from the reads, the theoretical maximum coverage is 85 – 87% as shown in the section on mapping techniques. We present a high confidence set of SNPs containing 353,151 positions, of which 6.8% are not reported in dbSNP [SWK+01]. Our analysis of contamination show a low level of around 0.8%.
Bibliography


Chapter 4

SNPest: Estimating genotype using a probabilistic graphical model

Stinus Lindgreen, Anders Krogh and Jakob Skou Pedersen

draft.
SNPest: Estimating genotype using a probabilistic graphical model

Stinus Lindgreen  Anders Krogh  Jakob Skou Pedersen

Bioinformatics Centre, Department of Biology, University of Copenhagen, Denmark

DRAFT

1 Abstract

Next generation sequencing techniques such as Illumina[3, 17], SOLiD[15] and 454[7] present researchers with the possibility of generating millions of short sequence reads within a short time frame. This development has revolutionized the field of genomics but also present researchers with new challenges that need to be addressed. To perform robust and high confidence genotyping and SNP calling, methods are needed that take the technology specific limitations into account and can model the different sources of error that are unavoidable in the data. We present the method SNPest which is a probabilistic generative model of the sampling and sequencing process. SNPest performs genotyping and SNP calling while explicitly taking advantage of the quality scores that accompany the read sequences. SNPest is also capable of modelling other factors that influence the result such as method or data specific errors. It is thus a flexible tool that can be used for many different platforms. We compare SNPest to the SAMtools suite of tools and also show the results of analysing a large dataset of ancient DNA extracted from an approximately 4000 years old human individual [11].

2 Introduction

There has been a revolutionizing development in sequencing technology from the first genome sequencing projects were initiated in the 1990s and up until today. Both the time frame and cost of sequencing have decreased significantly, and today a single research lab can generate millions of basepairs in a short time and for a reasonable price.

Next generation sequencing (NGS) techniques cover a wide range of technologies that succeed the Sanger sequencing approach [12]. A common feature in NGS is the high-throughput nature of the technique in which a large number of DNA templates is sequenced in parallel in a stepwise fashion [14, 8]. However, all these new sequencing techniques have their own method specific biases that need to be addressed in the subsequent analyses. 454 have problems with
homopolymers that will lead to insertion-deletion errors in the downstream analysis, while Illumina sequencing mainly have problems with substitution events and short read lengths due to degradation of the signal-to-noise ratio \[14\].

NGS methods generate a large amount of data but with various sources of error that need to be modeled for genotyping and SNP calling to work optimally. The SNP calling method has to be robust to noise and, preferably, not biased by the systematic errors in the NGS platform. Even if high-quality Sanger sequencing is used, some regions will have low coverage and so the need for careful modelling is still present.

A new but growing field is the sequencing of ancient DNA, such as the woolly mammoth \[9\] and the neandertal \[4, 10\]. This presents special challenges such as a limited amount of data and hence lower coverage and read depth, but also the fragmentation and damage suffered by the DNA over time poses some unique problems. Interestingly, novel medical applications present some of the same challenges as sequencing has been carried out on samples that have been formalin-fixed and paraffin-embedded \[13\].

In the widely used MAQ program \[6\] – now part of the SAMtools package \[5\] – a diploid genome is assumed, and the genotyping at a given position is based on the two most frequent nucleotides, \(a\) and \(b\). Thus, only three combinations are considered at each site: \(aa\), \(bb\) and \(ab\). Assuming a prior probability of a SNP to be \(r = 0.001\) by default, the prior for the two homozygotes become \(P(\text{aa}) = P(\text{bb}) = \frac{1 - r}{2}\). If available, MAQ also takes the quality of the mapping of each read into account. The posterior probability of each genotype is calculated, and the one with the highest posterior probability is reported.

We present the method SNPest, which models the genotyping and SNP calling from the raw read sequences in a fully probabilistic framework. The problem is described using a generative probabilistic graphical model \[1\]. There are many advantages in using a probabilistic model: The sampling and sequencing process is modelled explicitly making the approach flexible, all results get an intuitive confidence measure directly from the method, it can utilize all available information, and it is easily extended to take other sources of error or prior knowledge into account.

We have applied SNPest to the large dataset of the first human genome from an ancient individual \[11\]. The DNA was extracted from an approximately 4000 years old hair sample, and the data covers 79% of the genome, which is close to the theoretical maximum, at an average depth of 20X.

### 3 Method

When performing the mapping, the target genome will not be covered evenly by reads. Some regions will not be covered at all, while the read depth will vary from high to low in other parts. Furthermore, each position will not only be covered by nucleotides present at that position on the sampled individual, and the sampling will not always sample both alleles evenly (in case of a diploid genome). Wrongly mapped reads and errors in the base calling can also lead to a distribution of all nucleotides at a given position. In the case of ancient DNA, there is also the issue of DNA damage. All this adds noise to the picture making genotyping a hard problem.

Together with the reads the sequencing platforms generally provide a Phred-
like quality score [2] that indicates the probability of error at each position. These qualities \( Q \) are given as a sequence of ASCII characters, where the actual quality scores \( SC(Q) \) are offset by some value \( \Delta \), \( SC(Q) = ord(Q) - \Delta \), where \( ord \) gives the ASCII value of a character. The higher the quality score \( SC(Q) \), the lower the probability of an error:

\[
P_{\text{error}}(Q) = 10^{-SC(Q)}
\]

We developed SNPest (Single Nucleotide Polymorphism estimation) to be a sensitive genotyping tool, designed to avoid systematic biases due to e.g. read errors. SNPest is a probabilistic model that takes quality scores and alternative sources of errors explicitly into account. It is based on a generative model of the probability distribution over genotypes given the sampling and sequencing of nucleotides obtained from a diploid genome.

The graphical model is shown in Fig. 1 with circles representing random variables (RV). The two top RVs are global for a given position, whereas the boxed part of the model denotes the \( n \) individual reads covering the specific position. The observed RVs are the light intensities \( I_i \) and the reference nucleotide \( H \). The unobserved RVs are the nucleotide in the read sequence \( R_i \), the originally sampled nucleotide \( S_i \), and the genotype in the diploid genome \( G \). For a single position in the genome, the combined probability distribution can be written as:

\[
P(H,G,A,R,I) = P(H)P(G|H)\prod_{i=1}^{n}P(S_i|G)P(R_i|S_i)P(I_i|R_i)
\]

Figure 1: The graphical model used in SNPest. The two top RVs are global for a given position in the genome, while the RVs in the box model the contribution from each of the \( n \) reads covering the position.

The product is over the \( n \) reads with variables \( S_i, R_i \) and \( I_i \) corresponding to each of the reads. The probability factorizes into five separate probabilities: \( P(H) \) is the prior distribution over the reference nucleotide. \( P(G|H) \) is the conditional probability distribution over the ten possible genotypes \( \{AA, AC, \ldots TT\} \) given the reference nucleotide. \( P(S_i|G) \) is the conditional probability distribution over the actual nucleotide present in the sampled DNA fragment given the
genotype. $P(R_i|S_i)$ is the conditional probability distribution over the read nucleotides given the actual nucleotide. $P(I_i|R_i)$ is the conditional probability distribution over light intensities given the read nucleotide.

In our parameterization, a flat prior is used for $P(H)$ but it may be observed. The distribution $P(G|H)$ is estimated from the Yanhuang genome [18]. If the genotype is a homozygote in nucleotide $N$, then $P(S_i|G) = 100\%$ for nucleotide $N$ and $P(S_i|G) = 0\%$ for the three other nucleotides. If it is a heterozygote, the probability is $P(S_i|G) = 50\%$ for the two alleles and $P(S_i|G) = 0\%$ for the remaining two. The probability $P(R_i|S_i)$ models all sources of error from the original DNA to the sequencing step e.g. from damage, PCR amplification errors etc. The error rate was estimated to be 0.33\% by counting the observed number of mismatches from high quality read nucleotides. We do not know the form of the conditional probability distribution $P(I_i|R_i)$. However, this probability is proportional to $P(R_i|I_i)$ which is given by the quality scores.

A haploid version of SNPest is made by changing the distribution over the genotypes to $\{A,C,G,T\}$, and $P(S_i|G)$ is simplified to be the identity matrix.

SNPest calls the genotype with the highest posterior probability at each position. This probability is found by marginalizing the combined probability distribution and summing out the $S$ and $R$ parameters. This gives the probability $P(G, I, H)$, and by marginalizing $G$ we obtain $P(G|I, H)$. The posterior probability of the genotype reflects the confidence that SNPest has in inferring that particular genotype. This information can be used to generate a high confidence subset of genotypes if needed.

The observed random variable corresponding to $I$ is set to $P(R_i|I_i)$ which is given by the pairs of read nucleotides and quality scores. $P(H)$ is a flat prior over the four nucleotides that may be observed.

In SNPest, there are two parameters that can be tuned to fit the data. First, the prior on the genotype is estimated from a genome with a SNP frequency of 0.12\%. In case the data is known to have a higher or lower SNP frequency, the prior should be adjusted accordingly. The distribution over genotypes can be viewed as a $4 \times 10$ matrix with a row for each possible reference nucleotide. For a specific SNP frequency $F$, we calculate the parameter $\alpha$ that scales the frequencies of homozygotes:

$$\alpha = \frac{1 - F}{\sum_{N \in \{A,C,G,T\}} P(N)P(N,N)}$$

where $P(N)$ is the prior probability of nucleotide $N$ and $P(N,N)$ is the probability of the corresponding homozygote genotype. From this, $\beta$-parameters are calculated for each reference nucleotide $N$ to scale the heterozygote frequencies:

$$\beta_N = \frac{1 - \alpha P(N,N)}{\sum_{N' \neq N} P(N,N')}$$

where the sum is over nucleotides $N'$. In contrast, a completely flat prior over the 10 genotypes can also be used.

The other scaling parameter is used to model the errors in the data. Calculating the length 4 vector $\vec{P}_R$ describing the nucleotide distribution $P(R_i|I_i)$ directly from the quality scores will not consider e.g. wrong mappings or other sources of error. The expected error distribution is given by $P(R_i|S_i)$, and if
we expect an error rate of $\tau$, we multiply the vector $\vec{P}_R$ by an error matrix $A$, yielding the new probability vector $\vec{P}_A$:

$$\vec{P}_A = A \cdot \vec{P}_R$$

where

$$A = \begin{pmatrix}
(1-\tau) & \frac{\tau}{3} & \frac{\tau}{3} & \frac{\tau}{3} \\
\frac{\tau}{3} & (1-\tau) & \frac{\tau}{3} & \frac{\tau}{3} \\
\frac{\tau}{3} & \frac{\tau}{3} & (1-\tau) & \frac{\tau}{3} \\
\frac{\tau}{3} & \frac{\tau}{3} & \frac{\tau}{3} & (1-\tau)
\end{pmatrix}$$

By default, the error rate is estimated to be $\tau = 0.33\%$. The above error correction uniformly distributes the errors over all nucleotide probabilities. In the actual dataset used, some errors are much more common than others. By collecting high quality nucleotides with an error probability of $P_{\text{error}}(Q) \leq 0.04\%$, the number of discrepancies was counted and an error matrix estimated:

$$A_{\text{actual}} = \begin{pmatrix}
0.9982 & 0.0003 & 0.0010 & 0.0004 \\
0.0006 & 0.9979 & 0.0004 & 0.0011 \\
0.0010 & 0.0003 & 0.9980 & 0.0006 \\
0.0004 & 0.0010 & 0.0003 & 0.9983
\end{pmatrix}$$

4 Benchmarking

To analyze the effect of the tuning parameters, two test sets were generated as follows: From the large ancient DNA dataset mentioned, positions with a read depth between 80X and 100X were extracted. For each position, the frequency of each nucleotide was counted, and if a nucleotide $N$ was used at least 70% of the time, the correct genotype was set to be homozygote in $N$. If exactly two nucleotides were each used at least 30% of the time, the correct genotype was set to be heterozygote in the two nucleotides. The polymorphic positions – i.e. all heterozygote positions and all homozygote position where $N$ differs from the reference – were used as the positive set (23,483 positions), while a negative set was made from 100,000 randomly chosen positions from the rest. The expected SNP frequency is thus $F = 0.19$. The dependence on read depth is tested by, for each position, randomly sampling $D$ pairs of nucleotide and quality, $D = 1, 2, \ldots, 80$, and evaluating the performance.

The number of true positives ($TP$) is the number of positions in the positive set where the prediction by SNPest is the same as the correct genotype as defined above. The number of false negatives ($FN$) is the remaining positions in the positive set. Similarly, the true negatives ($TN$) are the positions in the negative set where SNPest predicted the correct (reference) nucleotide as the homozygote genotype. The false positives ($FP$) are the positions in the negative set called as SNPs by SNPest. The performance was evaluated using sensitivity ($SEN$) and specificity ($SPEC$):

$$SEN = \frac{TP}{TP + FN} \quad SPEC = \frac{TN}{TN + FP}$$

SNPest was used to call SNPs on both sets with different settings. In all cases, all genotypes from SNPest were trusted regardless of the posterior probability. The input was adjusted for different levels of error: The raw qualities ($\tau = 0\%$), using the estimated error rate of $\tau = 0.33\%$, and two higher levels of error of $\tau = 3\%$ and $\tau = 10\%$, respectively. In all cases, we assume a flat prior on the genotype (Fig. 2, top panel).
The estimated error rate of 0.33% is used in all subsequent calculations. The tuning of the genotype prior was tested by using a flat prior, the prior calculated from the Yanhuang genome [18] with a SNP frequency of 0.12%, and a modified prior with an expected SNP frequency of 20%. The 0.12% and 20% SNP frequencies were tested with and without the use of reference nucleotide information since this will affect the prediction. The flat prior is independent of the reference (Fig. 2, middle panel).

Using an expected SNP frequency of 20%, the influence of using the posterior probability when looking at the predictions is tested. When defining the true positives and true negatives, an additional check on whether the posterior probability is greater than or equal to the given threshold is carried out (Fig. 2, bottom panel).

The performance of SNPest was compared to that of SAMtools [5] on the testset described above along with the same metrics of quality. The test was carried out using SAMtools with the standard settings of an expected SNP frequency of 0.1% and a quality cutoff of 20, as well as with an expected SNP frequency of 20%. This is compared to SNPest using a threshold on the posterior probability of 90%, an error rate of 0.33% and with both an expected SNP frequency of 0.12% and 20% combined with either using or ignoring the reference nucleotide information (Fig. 3).

The comparison shows that for similar expected SNP frequencies, SNPest performs comparable to or slightly better than SAMtools. For both values of expected SNP frequency, SNPest outperforms SAMtools from a read depth of approximately 10X. At lower read depths, using the correct SNP frequency clearly affects the performance, but from a read depth of around 20X and up it does not seem to affect the result greatly since the information in the reads is enough to perform the correct call.

5 Performance on real-life data

SNPest has been used for the analysis of the ancient DNA dataset mentioned previously [11], retrieving the first ancient human genome with a high coverage of 79% and average read depth of 20X. The ancient DNA presented a number of challenges to overcome such as read errors and short genomic fragments. The short reads gave rise to inclusion of adapter sequence in the sequencing reaction, which again can lead to erroneous SNP calls if not properly dealt with. SNPest was used in the Saqqaq project as presented below.

Another problem that is encountered in many sequencing projects was clonal expansion where a few DNA fragments dominate the PCR amplification. These clonal reads are defined as having the same length, mapping to the same position on the same strand, and being from the same PCR library. When a "clone family" is discovered, all the reads are collapsed into a single representative sequence by combining the probabilities at each position. The most likely sequence from this new distribution is retrieved, and a new string of quality scores is calculated. The new error probabilities can be much lower than the minimum $10^{-4}$ in the Illumina pipeline. The range is extended by an additional 10 characters, allowing error probabilities down to $10^{-5}$. The range could be extended further, but to avoid biases from early errors that would be boosted in this process the range is truncated.
Figure 2: SNPest sensitivity (left) and specificity (right) as a function of read depth. Note different scales on Y-axes. **Top:** Error rates. Blue: Raw qualities. Red: $\tau = 0.33\%$. Green: $\tau = 3\%$. Yellow: $\tau = 10\%$. Purple: Actually observed error rates. **Middle:** SNP frequencies, using $\tau = 0.33\%$. Blue: Flat prior. Yellow: $F = 0.12\%$ using reference nucleotide. Light blue: $F = 0.12\%$ without reference. Green: $F = 20\%$ with reference. Red: $F = 20\%$ without reference. **Bottom:** Posterior probabilities, $F = 20\%$ and $\tau = 0.33\%$. The graphs go from including all positions (blue) to a posterior probability of at least 0.999 (orange).
SNPest was used on the genome wide Saqqaq data covering 2.4 billion base-pairs with an average read depth of 20X. Genotyping the entire dataset resulted in 2.2 million SNPs with an 86% overlap with dbSNP [16]. The SNP set was filtered to create a high confidence set of SNPs for further analysis. The filters applied were 1) read depth between 10X and 50X to avoid poorly covered and repetitive regions, respectively, 2) no overlap with annotated repeats, 3) a posterior probability of at least 0.9999, and 4) at least 5 nucleotides to the closest neighbouring SNP to avoid errors due to indels [18]. For the mitochondrial DNA with an average read depth of 3802X, we ignored the upper limit on read depth. The high confidence set contains 353,151 SNPs with an overlap with dbSNP of 93%.

When using the reference nucleotide in genotyping, the prior probability for a heterozygote genotype involving the reference will be higher than the prior probability for a homozygote call not involving the reference. This happens because the evidence in the observed reference would have to be outweighed by enough reads to completely ignore the reference. This is especially true for low read depths where the prior will dominate. Also, repetitive regions such as SINEs and LINEs can give rise to mapping errors as a read from one region in the genome will overlap with a SINE/LINE in many other positions. Similarly, repeats in unassembled regions give rise to biases, as reads from these regions will be mapped to assembled members of the repeat family in other parts of the genome. Indels can also give rise to wrong SNP calls as they are not modeled by most mapping tools and can hence introduce an offset in the mapping.

As a test of contamination, the haploid X chromosome from the male Saqqaq individual was also genotyped using the diploid model. This gave 76 high confidence heterozygote SNPs of which 29% are located within 10 nucleotides of indels or other structural variations. This is highly overrepresented compared to the 4.2% of chromosome X that are within this range of structural variation.
and indels. Another 24% were in regions with high sequence homology to other regions in the genome. The remaining heterozygote SNPs were either in regions with elevated SNP density, indicative of unannotated structural variation, or close to sequences with homologs throughout the genome. Thus, the majority of the heterozygote calls on chromosome X can be explained by mapping errors and structural variation.

6 Discussion

The developed SNPest program uses a generative probabilistic graphical model to address the problem of genotyping, taking various sources of error and information into account. The inferred genotype is presented with a posterior probability making it easy to generate high confidence SNP sets for sensitive analyses.

The probabilistic framework described above is intuitive and easily extended. For specific problems and datasets, the model can be designed to capture specific biases in the data or other aspects that should be considered in the genotyping and SNP calling. Although the focus here has been on Illumina sequencing data, SNPest can easily be adapted to other platforms and incorporate other sources of error. If SNPest is combined with a mapping tool assigning probabilities to the mapping of each read, the uncertainties can be directly incorporated into the model. Thus, SNPest is a versatile genotyping tool.

References


Chapter 5

Ancient Human Genome Sequence of an Extinct Palaeo-Eskimo

Morten Rasmussen, Yingrui Li, Stinus Lindgreen et al.

Ancient human genome sequence of an extinct Palaeo-Eskimo

Morten Rasmussen1,2*, Yingrui Li2,3*, Stinus Lindgreen1,2,*, Jakob Skou Pedersen4, Anders Albrechtsen4, Ida Moltke1, Mait Metspalu2, Ene Metspalu2, Toomas Kivisild2,3,*, Ramneek Gupta2, Marcelo Bertalan5, Kasper Nielsen2,*, M. Thomas P. Gilbert1,2, Yong Wang5, Maanasa Raghavan1,2,*, Paula F. Campos2, Hanne Munkholm Kamp2,1, Andrew S. Wilson1,4, Andrew Gledhill1,2, Silvana Trindaco1,12, Michael Bunce12, Eline Deirde Lorenzen1, Jonas Binladen5, Xiaosen Guo2,3, Jing Zhao2,3, Xiuxing Zhang2,3, Hao Zhang2,3, Zhuo Li3,*, Minfeng Chen2,3, Ludovico Orlando5,13, Karsten Kristiansen2,3,4,5, Mads Bak14, Niels Tommerup15, Christian Bendixen15, Tracey L. Pierre16, Bjarne Grønnov17, Morten Meldgaard18, Claus Andresen19, Sardana A. Fedorova20,21, Ludmila P. Osipova21, Thomas F. G. Higham22, Christopher B. Ramsey23, Thomas O. Hansen22, Finn C. Nielsen22, Michael H. Crawford23, Søren Brunak24,25, Thomas Sicheritz-Pontén1,*, Richard Villems2,3, Rasmus Nielsen4,8, Anders Krogh2,4, Jun Wang2,3,4,6, Eske Willerslev1,2

We report here the genome sequence of an ancient human. Obtained from ~4,000-year-old permafrost-preserved hair, the genome represents a male individual from the first known culture to settle in Greenland. Sequenced to an average depth of 20-fold, we recover 79% of the diploid genome, an amount close to the practical limit of current sequencing technologies. We identify 353,151 high-confidence single-nucleotide polymorphisms (SNPs), of which 6.8% have not been reported previously. We estimate raw read contamination to be no higher than 0.8%. We use functional SNP assessment to assign possible phenotypic characteristics of the individual that belonged to a culture whose location has yielded only trace human remains. We compare the high-confidence SNPs to those of contemporary populations to find the populations most closely related to the individual. This provides evidence for a migration from Siberia into the New World some 5,500 years ago, independent of that giving rise to the modern Native Americans and Inuit.

Recent advances in DNA sequencing technologies have initiated an era of personal genomics. Eight human genome sequences have been reported so far, for individuals with ancestry in three distinct geographical regions: a Yoruba African1,2, four Europeans3,4, a Han Chinese5, and two Koreans6,7, and soon this data set will expand significantly as the ‘1000 genomes’ project is completed. From an evolutionary perspective, however, modern genomics is restricted by not being able to uncover past human genetic diversity and composition directly. To access such data, ancient genomic degradation have also compromised the possibility of obtaining high sequence depth11, and no ancient nuclear genome has been sequenced deeper than about 0.7X—a level insufficient for genotyping and exclusion of errors owing to sequencing or post-mortem DNA damage11. In 2008 we used permafrost-preserved hair from one of the earliest individuals that settled in the New World Arctic (northern Alaska, and DNA from Greenland) belonging to the Saqqaq Culture (a component of the Arctic Small Tool tradition; approximately 4,750–2,500 14C years before present (yr BP))14–16 to generate the first complete ancient human mitochondrial DNA (mtDNA) genome16. A total of 80% of the recovered DNA was human, with no evidence of modern human contaminant DNA. Thus, the specimen is an excellent candidate upon which to sequence the first ancient human nuclear genome. Although

1,2 Centre for GeoGenetics, Natural History Museum of Denmark and Department of Biology, University of Copenhagen, Universitetsparken 15, DK-2100 Copenhagen, Denmark. 3Steno-Danish Genetics Centre, BGI-Shenzhen, Shenzhen 518083, China, and University of Copenhagen, DK-2900 Copenhagen, Denmark. 4BGI-Shenzhen, Shenzhen 518083, China. 5Department of Biology, University of Copenhagen, Ole Maaløes Vej 5, DK-2200 Copenhagen, Denmark. 6Department of Evolutionary Biology, Tartu University and Estonian Biocenter, 23 Ria Street, 51001 Tartu, Estonia. 7Laservitus Centre for Human Evolutionary Studies, Department of Biological Anthropology, Henry Wellcome Building, Fitzwilliam Street, University of Cambridge, Cambridge CB2 1QH, UK. 8Center for Biological Sequence Analysis, Department of Systems Biology, Technical University of Denmark, DK-2800 Lyngby, Denmark. 9Departments of Integrative Biology and Statistics, UC Berkeley, 4098 VLSB, Berkeley, California 94720, USA. 10Research Laboratory for Archaeology and the History of Art, Dyson Perrins Building, South Parks Road, Oxford OX1 3QY, UK. 11Department of Archaeological Sciences, School of Life Sciences, University of Bradford, West Yorkshire, Bradford BDD 1DP, UK. 12Biological Criministics, Australian Federal Police, 1 Union Place, Weston, ACT 2611, Australia. 13Ancient DNA Laboratory, School of Biological Sciences and Biotechnology, Murdoch University, Perth 6150, Australia. 14Paleogenetics and Molecular Evolution, Institut de Génomique Fonctionnelle de Lyon, Université de Lyon, Université Lyon 1, CNRS, INRA, Ecole Normale Supérieure de Lyon, 46 allée d’Italie, 69364 Lyon Cedex 07, France. 15Wilhelm Johannsen Centre For Functional Genome Research, University of Copenhagen, Department of Cellular and Molecular Medicine, The Panum Institute, Blegdamsvej 3A, DK-2200 Copenhagen, Denmark. 16Department of Genetics and Biotechnology, Aarhus University, Blichers Allé 20, FO BOX 50, DK-8830 Tjele, Denmark. 17Department of Biological Anthropology, University of Cambridge, Cambridge CB2 3QY, UK. 18Ethnographic Collections, National Museum of Denmark, Frederiksholms Kanal 12, DK-1220 Copenhagen, Denmark. 19Natural History Museum of Denmark, University of Copenhagen, Øster Voldgade 5-7, DK-1350 Copenhagen, Denmark. 20Greenland National Museum and Archives, PO Box 145, DK-3900 Nuuk, Greenland. 21Department of Molecular Genetics, Yakut Research Centre, Russian Academy of Medical Sciences, 4 Sergeyalakhonskoe Shosse, Yakutsk 677019, Sakha, Russia. 22The Institute of Cytology and Genetics, Siberian Branch of the Russian Academy of Sciences, Lavrentyeva Ave. Novosibirsk 630090, Russia. 23Department of Clinical Biochemistry, University of Copenhagen, 2200 Copenhagen, Denmark. 24Department of Anthropology, University of Kansas, Lawrence, Kansas 66045, USA. 25Novo Nordisk Foundation Center for Protein Research, Faculty of Health Sciences, University of Copenhagen, Blegdamsvej 3A, DK-2200 Copenhagen, Denmark.

*These authors contributed equally to this work.

Nature nature08835.3d 22/11/10 13:07:36
cultural artefacts from the Arctic Small Tool tradition are found many places in the New World Arctic, few human remains have been recovered. Thus, the sequencing project described here is a direct test of the extent to which ancient genomics can contribute knowledge about now-extinct cultures, from which little is known about their phenotypic traits, genetic origin and biological relationship to present-day populations.

2 Sample characteristics, DNA quality and sequencing strategy

The specimen used for genomic sequencing is the largest (approximately 15 x 10 cm) of four human hair tufts excavated directly from culturally deposited permafrozen sediments at Qeqertasussuk (Fig. 1a, b). Stable light isotope analyses of the Saqqaq hair (carbon and nitrogen) revealed that the individual relied on high trophic level marine food resources (Fig. 1e and Supplementary Information). Accelerator mass spectrometry (AMS) radiocarbon dating of the hair sample produced a date of 4,044 ± 31^{14}C yr BP and 4,170–3,600 cal yr BP when correcting for local marine reservoir effect (Supplementary Information). Despite its age, morphological analysis of the hair tuft using light and scanning electron microscopes indicated excellent overall preservation (Fig. 1c, d and Supplementary Information).

A major concern in ancient DNA studies is post-mortem damage, cytose to uracil deamination, that can result in erroneous base incorporation. Such miscoding lesions make it difficult to distinguish true evolutionarily derived substitutions from those that are damage-based, especially if sequence depth is low. It is therefore preferential to exclude damaged DNA molecules before sequencing, if achievable without loss of significant amounts of starting templates. We established the practical feasibility of this, by comparing Illumina sequencing libraries that were initially enriched using two different DNA polymerase enzymes: (1) Phusion polymerase (Finnzymes) as suggested in Illumina’s own library preparation protocol, which is not preferential to exclude damaged DNA molecules before sequencing, if achievable without loss of significant amounts of starting templates. We established the practical feasibility of this, by comparing Illumina sequencing libraries that were initially enriched using two different DNA polymerase enzymes: (1) Phusion polymerase (Finnzymes) as suggested in Illumina’s own library preparation protocol, which is not able to replicate through uracil (Supplementary Information). (2) Platinum Taq High Fidelity (HiFi, Invitrogen) polymerase, that can replicate through uracil (Supplementary Information).

Results allowed us to estimate an overall deamination-based damage rate of less than 1% in the Saqqaq genome. This ensures that any possible human contamination should reveal itself as being of European origin, given that any handling steps before indexing were carried out only by ethnic northern Europeans (Supplementary Information).

Sequencing and assembly

Twelve DNA libraries were built in the dedicated Copenhagen ancient DNA laboratory, several indexed enrichment PCRs were carried out, and each was sequenced on an average of three lanes using Illumina GAII sequencing platforms at BGI-Shenzhen. In addition, two sequencing runs were completed at Illumina’s facilities in Hayward, California and Chesterford, England. With few exceptions, 70 cycles of single-read sequencing were performed, always followed by a 6-bp index read (Supplementary Information). The sequencing yielded a total of 3.5 billion reads, from a total of 242 lanes.

Figure 1 | Sample details. a, Location of the Saqqaq Culture site Qeqertasussuk, north-western Greenland (after ref. 15). b, Saqqaq hair sample. c, Saqqaq and modern hair lengths on a comparison microscope. d, Comparative cross-sections of modern Caucasian and Saqqaq hairs. e, Carbon and nitrogen isotope measurements on the Saqqaq hair (brown square, Qt 86 profile C; 85/261:12 Oxford; pink triangle, Qt 86 profile C; 85/261:12 Bradford). Another Saqqaq hair sample from a similar context (green diamond, Qt 87 FB 20/20), six ancient Thule (Inuit) samples (purple circle). f, Calibrated ages (before present) on the Saqqaq hair and associated reindeer bones, plotted using the INTCAL04 calibration curve, are shown. The human hair dates are calibrated twice, once using an correction for the marine reservoir effect (Supplementary Information).
Sequences not carrying a 100% match in the index read were excluded from all downstream analyses. This allowed 93.17% of all reads to be attempted to be mapped to the human reference genome (hg18) using a suffix array-based mapping strategy that permits identification of residual primer sequence expected from the libraries of short ancient DNA fragments (Supplementary Information). Primer trimming was carried out as an integrated part of the mapping during the alignment of each read to the genome. Specifically, for all positions a check was made as to whether a better alignment could be made between the remainder of the read and the primer. If found, this position in the read was used to cut off the primer (Supplementary Information). This provided an average mapped read length of 55.27 nucleotides. Of the correctly indexed reads, 49.2% could be mapped uniquely (46% of total reads). Reads with multiple matches or no matches were discarded (Fig. 2a). Analysis of the reads with no matches indicated that most were unidentifiable, whereas the remainder were of microbial eukaryote, viral, or bacterial origin (Fig. 2b). Read sequences from the same library that were mapped to the reference genome with same start and end positions were considered clonal, and were collapsed to single sequences with higher scores and different sources of read errors into account. For the sex chromosomes, we developed a probabilistic model of the sampling of reads from the diploid genome, called SN Pest, which takes quality scores and different sources of read errors into account. For the sex chromosomes and the mtDNA a haploid model was used. Given the

mapped reads and their quality scores, we assigned the most probable genotype to each position (Supplementary Information). We performed genotyping on all positions, using all available read information for depths ≤200×. For read depths >200×, we based the genotyping on 200 randomly sampled reads. This simplification was shown to have negligible effect on the results while speeding up the calculations markedly (Supplementary Information). This resulted in 2.2 million SNPs (Fig. 2a), of which 86.2% have previously been reported (dbSNPv130).

We additionally defined a high-quality subset of SNPs, based on positions with read depth between 10× and 50×, to avoid poorly covered and repetitive regions with extreme read depth. We also demanded that these SNPs have posterior probabilities of >0.9999, not to be positioned in annotated repeat regions, and to have a distance of at least 5 bp to the closest neighbouring SNP to account for insertion and/or deletion (indel) errors. This provided a total of 353,151 SNPs with a 93.2% overlap with dbSNP (v130) (Fig. 2a).

The mtDNA genome was sequenced to an average depth of 3,802×. The consensus was identical to that previously recovered by GS FLX sequencing, except that a single position previously called as a heterozygote was now called as a C. Using the diploid model, no high-confidence heterozygotes were found. Applying the diploid model to the X chromosome resulted in 1,707 homozygote (versus 3,071 with the haploid model) and 76 heterozygote high-confidence SNPs. Of the latter, 29% can be explained by known indels and structural variation, whereas the remaining can be referred to mapping errors in repetitive regions (Supplementary Information). For the Saqqaq Y chromosome, we found 23 homozygote (versus 243 with the haploid model) and 445 heterozygote high-confidence SNPs. We explain the latter by the well-known fact that human Y chromosomes are difficult to assemble due to structural and repetitive regions. Importantly, the number of heterozygote SNPs found in the X and Y chromosomes when changing to the diplod model are similar to those from modern human genome sequencing (Supplementary Information).

Assessing contamination using the frequency of private European alleles (as defined in the human genome diversity project) as an estimator and a fixed error rate from the observed neighbouring bases, we estimate the raw read contamination to be at most 0.8% (standard error (s.e.) ≥ 0.2%) (Supplementary Information), a level comparable to those from modern human genome sequencing (Supplementary Information). We estimate that it is theoretically possible to cover some 85–87% of the genome (Supplementary Information), meaning that we are close to the reference genome with same start and end positions were considered clonal, and were collapsed to single sequences with higher scores and different sources of read errors into account. For the sex chromosomes, we developed a probabilistic model of the sampling of reads from the diploid genome, called SN Pest, which takes quality scores and different sources of read errors into account. For the sex chromosomes and the mtDNA a haploid model was used. Given the

mapped reads and their quality scores, we assigned the most probable genotype to each position (Supplementary Information). We performed genotyping on all positions, using all available read information for depths ≤200×. For read depths >200×, we based the genotyping on 200 randomly sampled reads. This simplification was shown to have negligible effect on the results while speeding up the calculations markedly (Supplementary Information). This resulted in 2.2 million SNPs (Fig. 2a), of which 86.2% have previously been reported (dbSNPv130).

We additionally defined a high-quality subset of SNPs, based on positions with read depth between 10× and 50×, to avoid poorly covered and repetitive regions with extreme read depth. We also demanded that these SNPs have posterior probabilities of >0.9999, not to be positioned in annotated repeat regions, and to have a distance of at least 5 bp to the closest neighbouring SNP to account for insertion and/or deletion (indel) errors. This provided a total of 353,151 SNPs with a 93.2% overlap with dbSNP (v130) (Fig. 2a).

The mtDNA genome was sequenced to an average depth of 3,802×. The consensus was identical to that previously recovered by GS FLX sequencing, except that a single position previously called as a heterozygote was now called as a C. Using the diploid model, no high-confidence heterozygotes were found. Applying the diploid model to the X chromosome resulted in 1,707 homozygote (versus 3,071 with the haploid model) and 76 heterozygote high-confidence SNPs. Of the latter, 29% can be explained by known indels and structural variation, whereas the remaining can be referred to mapping errors in repetitive regions (Supplementary Information). For the Saqqaq Y chromosome, we found 23 homozygote (versus 243 with the haploid model) and 445 heterozygote high-confidence SNPs. We explain the latter by the well-known fact that human Y chromosomes are difficult to assemble due to structural and repetitive regions. Importantly, the number of heterozygote SNPs found in the X and Y chromosomes when changing to the diplod model are similar to those from modern human genome sequencing (Supplementary Information).

Assessing contamination using the frequency of private European alleles (as defined in the human genome diversity project) as an estimator and a fixed error rate from the observed neighbouring bases, we estimate the raw read contamination to be at most 0.8% (standard error (s.e.) ≥ 0.2%) (Supplementary Information), a level comparable to those from modern human genome sequencing (Supplementary Information).
that will not affect our high-confidence genotype calls and will have a negligible effect otherwise. We investigated the Saqqaq individual for signs of inbreeding using two new statistical approaches that circumvent the problem of uncertainty in the genotype calls of heterozygotes, using the Siberian populations from Supplementary Table 12 as a reference. The methods provide a genome-wide estimate of the inbreeding coefficient \( F \) and identify regions of identity by descent (IBD) across the genome (Supplementary Fig. 13). The estimated value of \( F \) is 0.06 (s.e. 0.011) assuming no genotyping errors, which is equivalent to an offspring of two first cousins, but could have been caused by other family relationships of the parents (Supplementary Information). A positive value of \( F \) could possibly also be explained by population subdivision between the Saqqaq population and the Siberian reference population, or by natural selection. However, as many IBD tracts are \( >10\) Mb, far longer than the extent of linkage disequilibrium in the human genome, inbreeding within the Saqqaq population is more likely.

**Functional SNP assessment**

Although the relationship between risk allele and causation is still in its infancy\(^1\), some phenotypic traits can possibly be inferred from the genome data (all functional SNPs discussed below are listed in Supplementary Table 14). We only included genotypes with a posterior probability above 99%.

Given the A1 antigen allele plus encoding of the rhesus factor in combination with lack of B antigen and the O antigen frameshift mutation, we conclude that the Saqqaq individual had blood type \( A^\text{1/2} \). Although common in all ethnic groups, this has very high frequencies in populations of the east coast of Siberia down to mid China\(^2\). Furthermore, we find a combination of four SNPs at the \( HERC2 \)-\( OC2A \) locus, which among Asians is strongly associated with brown eyes\(^3\). SNPs on chromosomes 2, 5, 15 and X suggest that he probably did not have a European light skin colour\(^4\), had dark and thick hair\(^5\) (in agreement with the morphological examination (Fig. 1b–d)), and an increased risk of baldness\(^6,7\). The same SNP that is characteristic of hair thickness also suggests that he probably had shovel-graded front teeth—a characteristic trait of Asian and Native American populations\(^8\). An AA genotype SNP (forward strand) on chromosome 16 is consistent with the Saqqaq individual having earwax of the dry type that is typical of Asians and Native Americans, rather than the wet earwax type dominant in other ethnic groups\(^9\). In addition, the combined influence of 12 SNPs on metabolism and body mass index indicate that the Saqqaq individual was adapted to a cold climate (see Supplementary Information and Supplementary Table 14).

**Population genetics context of the Saqqaq individual**

The origin of the Saqqaq and other Paleo-Eskimo cultures, and their relationship to present-day populations, has been debated since they were first discovered in the 1950s\(^{10}\). Competing theories have attributed the origins to offshoots of the populations that gave rise to Native American populations such as the Na-Dene of North America, alternatively from the same source as the Inuit currently inhabiting the New World Arctic, or from still other sources entering the New World even later than both the Native American and Inuit ancestors (for summary see ref. 35).

A recent SNP genotyping study\(^{11}\) of the HGDP-CEPH panel of 51 populations has provided comprehensive global coverage of modern human genomic variation, but is limited with respect to Arctic populations. Therefore, we carried out Illumina Bead-Array-based genotyping on four native North American and twelve north Asian populations (Supplementary Table 12). A total of 95,502 SNPs from the resulting combined data set of 35 Eurasian and American populations was covered by high-quality data in the Saqqaq genome and was subject to further analyses (Fig. 3a–c and below).

Principal component analysis (PCA) was used to capture genetic variation. PC1 distinguishes west Eurasians from east Asians and Native Americans, whereas the PC2 captures differentiation between native Asians and Americans (Fig. 3b). Interestingly, the PC1 versus PC2 plot shows that the Saqqaq individual falls in the vicinity of three Old World Arctic populations—Ngonassans, Koryaks and Chukchis, while being more distantly related to the New World groups (Amerinds, Na-Dene and Greenland Inuit). Koryaks and Chukchis are in agreement with the PCA plots and suggest shared common influences, specifically the ones characteristic of native populations in East Asia, Siberia in particular, and the Arctic, on both sides of the Bering Strait (Fig. 3c). In this respect the populations closest to the Saqqaq are Koryaks and Chukchis. Importantly, in contrast to Saqqaq and Koryaks, modern Greenlanders carry clear evidence of admixture or shared ancestry with Amerindians. Moreover, at \( K = 5 \), the Inuit do not display genetic components of Siberians other than the ‘Beringian’ seen in Chukchis and Koryaks. The admixture results are in agreement with the PCA plots and suggest shared common ancestry of Saqqaq and modern Inuit before the movement of the former to the New World.

We additionally used a population genetic model to obtain maximum likelihood estimates of the divergence times between the Saqqaq individual and the reference populations (Supplementary Information). The population with the shortest divergence time was Chukchis, with an estimated divergence time of approximately 7,000 years \((0.043 \pm 0.008) N_e \) generations, where \( N_e \) is the effective population size. In contrast, the estimated divergence times to the other closely related populations—Na-Dene, Koryaks and Ngonassans—were 0.093, 0.11 and 0.089, respectively. The estimated divergence time to the Han Chinese, a more distantly related population, was 0.25. These estimates can be converted to estimates of years or generations, by making assumptions regarding the effective population sizes of the reference populations. The effective population sizes are in general unknown, but can be estimated from DNA sequence data, and are generally much smaller than the census sizes (Supplementary Information).
Information). We found no evidence in favour of changes in population size. Even when accounting for the uncertainty in the estimate of the mtDNA mutation rate, and possible biases related to the genotyping data, it is still unlikely that N0 > 5,000, providing a maximal divergence time between Chukchis and Saqqâqs of 175–255 generations or between 4,400 and 6,400 years. The oldest archaeological evidence of the Arctic Small Tool tradition in the New World is from Kuzitirin Lake, Alaska, dating back ~5,500 cal. yr BP[33], indicating that the ancestral Saqqâq separated from their Old World relatives almost immediately before their migration into the New World.

Conclusion

We report the successful genome sequencing of a ~4,000-year-old human. Data authenticity is supported by: (1) the private SNP analyses that indicate contamination levels in the raw sequence data to be ≤0.8%; (2) the mtDNA and Y-chromosome DNA haplotypes fit within haplogroups typical of north-east Asia; (3) population admixture analyses do not record any European component in the Saqqâq genome; and (4) the PCA plots clearly reveal close affiliation of the Saqqâq genome to those of contemporary north-east Siberian populations. These observations, coupled with evidence of excellent DNA preservation, and sample handling being restricted to northern Europeans before incorporation of a sequence indexing, indicate that contamination in the Saqqâq genome is not of concern. Our study thus demonstrates that it is possible to sequence the genome of an ancient human to a level that allows for SNP and population analyses to take place. It also reveals that such genomic data can be used to identify important phenotypic traits of an individual from an extinct culture that left only minor morphological information behind. Additionally, the ancient genomic data prove important in addressing past demographic history by unambiguously showing close relationships between Saqqâq and Old World Arctic populations (Ngnassans, Koryaks and Chukchis). A single individual may, or may not, be representative of the extinct culture that inhabited Greenland some 4,000 yr BP. Nevertheless, we may conclude that he, and perhaps the group that once crossed the Bering Strait, did this independently from the ancestors of present-day Native Americans and Inuit, and that he shares ancestry with Arctic north-east Asians, genetic structure components of which can be identified in many of the present-day people on both sides of the Bering Sea. The next technical challenge will be to sequence an ancient human genome from material outside the permafrost regions. Although undoubtedly challenging, it will, if successful, take the emerging field of palaeogenomics to yet another level.

METHODS SUMMARY

DNA was extracted from a ~4,000-year-old hair sample recovered from Qeqertarsuup, Greenland. Indexed Illumina libraries were sequenced following the manufacturer’s protocol, and images processed using pipeline v1.4. Reads with correct index were mapped to the human genome (hg18) with a suffix array-based method that allows for residual primer trimming (Supplementary Information). Genotyping was carried out using a probabilistic model, SNPess, designed to take into account errors specific for ancient samples (Supplementary Information).

Figure 3 | Population genetics and phylogenetics. a. Locations of the studied populations are shown with the most relevant populations indicated by name (numbers in circles correspond to the nr column in Supplementary Table 12). b. PCA plot (PC1 versus PC2) of the studied populations and the Saqqâq genome. c. Ancestry proportions of the studied 492 individuals from 35 extant American and Eurasian populations and the Saqqâq individual as revealed by the ADMIXTURE program™ with K = 5. Each individual is represented by a stacked column of the five proportions, with fractions indicated on the y axis. The analysis assumes no grouping information. The samples are sorted by region/population only after the analysis. For better readability the Saqqâq individual is shown in three columns. Populations added to the published collection[38] are shown in semi-bold. Red dots in the expanded plot indicate four individuals whose ancestry proportion pattern showed the highest correlation (Kendall τ > 0.95; P < 0.05) with that of the Saqqâq individual. d. The phylogenetic tree of Y chromosome haplogroup Q. The position of the Saqqâq individual is ascertained by markers shown on the tree. Information for markers shown in parentheses is missing and their status is therefore inferred. Haplogroup names are according to ref. 38; hash symbol indicates error in reference (Supplementary Information).

Nature nature08835.3d
Received 30 November 2009; accepted 18 January 2010.


Supplementary Information is linked to the online version of the paper at www.nature.com/nature.

Acknowledgements Centre for Geogenetics, the Copenhagen branch of the Sino-Danish Genomic Centre and Wilhelm Johansen Centre for Functional Genome Research were supported by Danish National Science Foundation, the Lundbeck Foundation, and the Danish Agency for Science, Technology and Innovation. Center for Biological Sequence Analysis was supported by Villum Kan Rasmussen Fonden; Center for Protein Research by the Novo Nordisk Foundation. E.W. thanks F. Paulsen for financial support to initiate the project. E.M. thanks Estonian Science Foundation for grant 7858, and R.V. EC DG for FP7 Eugene grant 205419 and EU RDF through Centre of Excellence in Genomics grant. J.W. thanks the Shenzhen Municipal Government, the Yantian District local government of Shenzhen, the National Natural Science Foundation of China (30725008), Ole Romer grant from the Danish National Science Research Council, the Solaex project (277-07-0196), and Danish Strategic Research Council (2006-07-0021). M. Bunce acknowledges the support of the Australian Research Council. A.K., S.L. and H.M.K., were supported by a grant from the Novo Nordisk Foundation and J.S.P. The Danish Council for Independent Research Medical Sciences. M.H.C. thanks the National Science Foundation for support of the Aleutian and Siberian projects through grants NSF OPP-9905909 and OPP-0327676. We thank G. Hudjashov, M. Neils, L. Anton, V. Sos, A. Wessolowa, H. H. Staerfeld, K. Kasrack, P. Wad Sackett, J. Li, H. Yu, Y. Huang, H. Zheng, H. Liang and T. Brand for technical help and Biobase for curation of selected phenotypes and Illumina’s core facilities in Hayward, California and Chesterfield, England. Use of HGMPD Professional was licensed through DTV at the Technical University of Denmark.


Author Information Sequences have been deposited to the short read archive with accession number SRA010102; summary data are also available via http://www.anzientogene.dk. Reprints and permissions information is available at www.nature.com/reprints. The authors declare no competing financial interests. This paper is distributed under the terms of the Creative Commons Attribution-Non-Commercial-Share-Alike license, and is freely available to all readers at www.nature.com/nature. Correspondence and requests for materials should be addressed to E.W. (ewillerslev@snm.ku.dk) or J.W. (wangj@genomics.org.cn).
Concluding Remarks

In this dissertation, I have presented the work I have done as a Ph.D student at the Bioinformatics Center. My work has been divided into two distinct subject areas. However, the one thing tying them together is also perhaps the most broadly applicable area of bioinformatics – namely sequence alignment. Within each field, though, there is a thread of continuity.

The first part of my work has been focused on RNA secondary structure prediction and alignment of multiple RNA sequences. The three first author papers presented in this part of my dissertation follow a logical development: The first paper in chapter 1 describes our work on developing measures of covariation that can be used to better predict basepairs in multiple alignments of RNA sequences. The goal here was to use the evolutionary signal present in the alignments to find alignment columns that support basepairing interactions. This work is of relevance for researchers working on methods for comparative structure prediction of RNAs.

The second paper in chapter 2 presents the method MASTR that we have developed for simultaneous multiple alignment and secondary structure prediction of RNA sequences. This program uses sampling guided by simulated annealing, thus avoiding the problem of local minima in the energy landscape. Furthermore, we use the results from the first paper to predict the basepairs in MASTR, tying the two projects together. One of the forces of MASTR is that it can handle multiple sequences and tries to handle the two problems of alignment and structure in a combined fashion.

The third paper in chapter 3 is on a higher level of abstraction and presents the meta-server WAR for aligning and predicting the structure of a set of RNA sequences using various methods. After working with different structure prediction methods for some time, you realize that we probably will not find the single program that always performs best – rather, different programs have different strengths and weaknesses, and one should use the best program for the problem at hand. However, this can be hard to know in advance. Our goal with developing the webserver WAR was to make it easy to use different methods on your dataset and at the same time get an overview of the performance.

For the RNA part of my work, there is a path of development from measures of
covariation over method development to a meta–server combining various approaches. From a user point-of-view, WAR will probably be of most use.

A possibly interesting development of MASTR would be to make it useful for the new genomic data. As sequencing techniques keep improving, the number of sequenced genomes grows rapidly making comparative prediction of conserved structural RNAs a highly relevant area of research. Other programs exist for this particular problem although it has not been the focus of my work and, hence, of this dissertation. However, extending MASTR to perform local alignment would make it usable for scanning multiple alignments for conserved structural RNAs.

The second part of my work took a completely different focus and forced me to become acquainted with a new field, namely next generation sequencing as specifically applied to ancient DNA. This is a very active field of research at the moment as the technology evolves at a rapid pace. The goal of this work was to generate the first genome of an ancient human individual.

One of the developments in our group was the program SNPest for performing genotyping and SNP calling. The generative approach taken using a probabilistic graphical model makes it possible to model different sources of error, making it useful for not only ancient DNA but sequencing projects in general. In the draft paper presented in chapter 4, we present the method in some detail along with a comparison to the method used on the large 1000 genomes project.

The sequencing project utilized the Illumina platform on ancient DNA extracted from 4000 years old hair, which presented additional interesting problems to solve. Our work for the project was method development, data handling, as well as the actual mapping of the genome and SNP calling for the analysis. A project this size involves a lot of smaller sub-projects, but the main accomplishment is presented in the paper in chapter 5, where we present the first genome sequence of an ancient human individual.

This is a very active area of research. The Saqqaq project presents the first large scale extraction of genomic DNA preserved in hair for thousands of years, and this can be extended to other preserved specimens. Just as the Saqqaq project presents evidence for a migration into the New World separate from the one that gave rise to the Inuit and modern Native Americans, similar genomic studies of mummified remains can help answer questions about e.g. population movements from the Old World into the Americas. In this process, SNPest can also be extended to directly incorporate damage information in the genotyping.

My research for the past three years has thus ranged from RNA to ancient DNA with various milestones along the way. Although there is not one red thread that ties it all together, my overall focus has been on the algorithmics and method development for the problem at hand. Both areas are rapidly evolving with interesting perspectives for further research. Presently, I feel that the sequencing projects are very exciting, and the first steps I have taken in this direction have only scratched the surface.
I would like to thank all the people who I have been working with or who have in other ways been involved for the past 3 years. First and foremost, my supervisor Anders Krogh who first introduced me to the field of bioinformatics as a Master’s student on the newly started program and since hired me as a Ph.D student and gave me the opportunity to teach others about the wonders of bioinformatics.

Most of this work is also greatly inspired by my friend and co-supervisor Paul Gardner, who has since left Denmark and now lives in some British town called Cambridge. I owe a great deal to the honorable Dr. Gardner.

I have been sharing my office with Jakob Skou Pedersen for some time, and we have been working together on the Saqqaq project. I would like to thank him for many academic discussions and conversations in general.

When I started, Morten Lindow was finishing his Ph.D in bioinformatics. We have been planning and teaching the course “Introduction to Bioinformatics” together which has been a lot of fun.

Elfar Torrarínsson and I started as Master students together and have been working on a number projects over the years. I also want to thank Troels Marstrand for our many conversations (sometimes even scientific) and for his comments on the manuscript. Eivind Valen has taught me that Norwegians can be ok.

I also want to thank Thomas Hamelryck for our conversations on various obscure subjects with or without an accompanying beer.

Morten Rasmussen for all the conversations and discussion we had while working on the Saqqaq genome project, and Eske Willerslev for asking me to be part of that project to begin with. It was a lot of work but all worth it in the end.

Hanne Munkholm also deserves special notice for keeping the system running, for being of great help during the Saqqaq project, and for mostly having time for a quick piece of cake during the afternoon.

Last but definitely not least, my wife Mirjam for putting up with me - even when times were stressed - and for reminding me to take some time off to enjoy our son, Thore, once in a while.

**Funding** The first part of my research was funded by a Novo Scholarship. The rest has been funded by a faculty stipend.
Curriculum Vitae

Bioinformatics Centre
Department of Biology
University of Copenhagen
Ole Maaloes Vej 5
2200 Copenhagen N
DENMARK
(+45) 3532 2187
stinus@binfo.ku.dk
http://people.binfo.ku.dk/stinus/

PERSONAL DETAILS

- **Full name:** Stinus Lindgreen
- **Date of Birth:** 9 August 1980
- **Citizenship:** Danish
- **Marital status:** Married
- **Language skills:** Danish (mother tongue), English (excellent)

EDUCATION

- **M.Sc., bioinformatics,** graduated: January 2006
  “Simultaneous Multiple Alignment and Structure Prediction of RNA Sequences”
  University of Copenhagen, Denmark
- **B.Sc., computer science,** graduated: May 2003
  “Simultaneous Multiple Alignment and Phylogenetic Inference”
  University of Copenhagen, Denmark
- **High school, mathematical-scientific path,** graduated: June 1999
  Gladsaxe Gymnasium, Denmark
Scholarships

- **Travel Grant**, Novozymes, 2006
- **IMA funding**, participation in IMA meeting: "RNA in Biology, Bioengineering and Nanotechnology", University of Minnesota, 2007
- **Travel Grant**, Oticon, 2008

Research Interests

- **Bioinformatics**: RNA secondary structure prediction, multiple sequence alignment, phylogenetic inference, sequence assembly and mapping, genotyping.
- **Computer science**: Algorithmics, NP completeness, time complexity, advanced heuristics, Self–Organizing Maps, Markov Chain Monte Carlo, simulated annealing.

Skills

- Experienced programmer in C, C++, Java, SML, Python, Bash and Shell scripting
- Some experience with PERL, Prolog and FORTRAN
- Experience with mathematical modelling
- Experienced user of R
PUBLICATIONS INCLUDED IN THESIS

  Measuring covariation in RNA alignments: Physical realism improves information measures
  Bioinformatics, 22(24):2988-2995

  MASTR: Multiple alignment and structure prediction of non-coding RNAs using simulated annealing
  Bioinformatics, 23(24):3304-3311

- **E. Torarinsson** and **S. Lindgreen** (2008)
  WAR: Webserver for aligning structural RNAs
  Nucleic Acids Research (Webserver issue), doi:10.1093/nar/gkn275
  *The two authors should be considered as joint first author*

- **M. Rasmussen**, L. Yingrui*, **S. Lindgreen** et al. (2009)
  Ancient human genome sequence of an extinct Palaeo-Eskimo
  Nature (in press)
  *The three authors should be considered as joint first author*

PUBLICATIONS NOT INCLUDED IN THESIS

  Deferred path heuristic for phylogenetic trees revisited
  In M.-F. Sagot and K. S. Guimarães (eds), CompBioNets 2005: Algorithmics and Computational Methods for Biochemical and Evolutionary Networks, volume 5 of Texts in Algorithmics, 75–92, King’s College London, Strand, London

  Multiple alignment and structure prediction of non-coding RNA sequences (poster abstract)
  BMC Bioinformatics, 8(Suppl 8):P8 from the Third ISCB Student Council Symposium at the Fifteenth Annual International ISMB Conference

  Rfam: updates to the RNA families database
  Nucleic Acids Research (Database issue), D136-40
CONTRIBUTIONS TO THE RESEARCH COMMUNITY

- **Referee:** Reviewer for ISMB, RECOMB, Bioinformatics, NAR, AMB and IEEE/ACM.
- **Poster:** “Simultaneous structure prediction and multiple alignment of RNA sequences”, BioSys meeting, Copenhagen, 2005
- **Poster:** “Structural multiple alignment of RNA using MCMC”, Evolution of Biomolecular Structure, Vienna, 2006
- **Poster:** “Structural multiple alignment of RNA using MCMC”, Bioinformatics 2006, Aarhus, 2006
- **Poster:** “Multiple alignment and structure prediction of non-coding RNA sequences”, ISMB/ECCB, Vienna, 2007
- **Poster:** “MASTR: Simultaneous multiple alignment and structure prediction of non-coding RNAs using simulated annealing”, RNA in Biology, Bioengineering and Nanotechnology, Minneapolis, MA, 2007
- **Presentation:** “Simultaneous structure prediction and multiple alignment of RNA sequences”, Novo Scholarship Symposium, 2006
- **Presentation:** “Structural Alignment of RNA: An MCMC Approach”, Copenhagen RNA-informatics Symposium, 2006
- **Presentation:** “WAR: Webserver for Aligning Structural RNAs”, Benasque, 2009
- **Workshop:** “Computational approaches to functional and regulatory RNAs”, Benasque, 2006
- **Workshop:** “Computational Methods for RNA Analysis”, Benasque, 2009
WORK EXPERIENCE

○ **Ph.D student**, Bioinformatics Centre, University of Copenhagen, Copenhagen, Denmark (November 2006 – January 2010)
  · Teaching introductory bioinformatics course, working on RNA secondary structure prediction and multiple alignment, performing data analysis and method development for genome assembly.

○ **Research assistant**, Bioinformatics Centre, University of Copenhagen, Copenhagen, Denmark (January 2006 – October 2006)
  · Worked on RNA secondary structure prediction and related topics.

○ **Teacher**, Bioinformatics Centre, University of Copenhagen, Copenhagen, Denmark (February–April 2006, February–April 2007, February–April 2008, February–April 2009)
  · Organized and taught the course “Introduction to bioinformatics” for undergraduate students in biology and biochemistry.

○ **Board member**, Bryggeriet Djævlebryg, Copenhagen, Denmark (May 2006–present)
  · Co–founder of the microbrewery *Bryggeriet Djævlebryg Aps* (www.djaevlebryg.dk).

○ **Board member**, Danish Atheist Society, Denmark (April 2008–present)
  · Part of the board in the Danish Atheist Society (www.ateist.dk)

○ **Council member**, Atheist Alliance International (October 2009–present)
  · In charge of international affairs, organizing Copenhagen conference (www.atheistalliance.org)

○ **Board member**, study board for bioinformatics, University of Copenhagen, Copenhagen, Denmark (2004–2005)
  · Helped organize the Master’s programme in Bioinformatics, structure the courses, go through feedback from students as well as review applicants.

○ **Instructor**, Department of Computer Science, University of Copenhagen, Copenhagen, Denmark (2003–2004)
  · Instructor in both basic and advanced algorithmics courses as a supplement to the lectures. Responsible for exercises, reports, homework assignments and general questions.

  · Taught basic programming at an evening school.

○ **Editor**, Danish Atheist Society newsletter, Denmark (May 2007–present)
  · Editor of the bi–monthly newsletter from the Danish Atheist Society (www.ateist.dk)