Predicting RNA Secondary Structures with Pseudoknots by MCMC Sampling.

— preprint —

Dirk Metzler* Markus Nebel†

June 19, 2006

Abstract

The most probable secondary structure of an RNA molecule, given the nucleotide sequence, can be computed efficiently if a stochastic context-free grammar (SCFG) is used as the prior distribution of the secondary structure. The structures of some RNA molecules contain so-called pseudoknots. Allowing all possible configurations of pseudoknots is not compatible with context-free grammar models and makes the search for an optimal secondary structure NP-complete.

We suggest a probabilistic model for RNA secondary structures with pseudoknots and present a Markov-chain Monte-Carlo Method for sampling RNA structures according to their posterior distribution for a given sequence. We demonstrate the benefit of our method in examples with tmRNA and simulated data.

1 Introduction

For many RNA molecules it is not the sequence of nucleotides (primary structure) that is of paramount importance but the molecule's conformation in space (tertiary structure). Since lab techniques for determining the tertiary structure are both expensive and error-prone it is a long-time goal of research to find reliable algorithms that derive a molecule's conformation from its primary structure. Because of the computational complexity of this task and the limited performance of older computers first of all attention has been paid to the prediction of a molecule's

*Institut für Informatik, J. W. Goethe-Universität, Frankfurt am Main, Germany, metzler@cs.uni-frankfurt.de, www.cs.uni-frankfurt.de/∼metzler

†Fachbereich Informatik, Universität Kaiserslautern, Germany

secondary structure. There the conformation is restricted to stay planar, additionally, so-called pseudoknots are prohibited. So, from a mathematical point of view the secondary structure becomes a special kind of planar graph as introduced by Waterman in 1978 [29]. The first algorithm for predicting the RNA secondary structure was proposed by Nussinov et al. [24]. Using a dynamic programming approach, Nussinov's algorithm finds one secondary structure with a maximal number of paired bases (assuming this to approximate a conformation of minimal free energy). Of course, this does not lead to reliable predictions. However, Nussinov's approach is still the core of some algorithms like e.g. the Zuker algorithm [33] that nowadays are used successfully. Major progress has been achieved by Zuker using a more accurate model for the free energy of a molecule. The model used can be traced back to Tinoco et al. [27] where one assumes that the free energy of a structure behaves like the sum of the free energies of all its loops. Using this energy model and searching for a secondary structure of minimal free energy (still using Nussinov's dynamic programming scheme) led to much better results, and in general refinements of the model lead to more accurate predictions.

A second class of algorithms is based on stochastic models of the molecule's structure. Examining a set of known molecule foldings it becomes possible to derive the maximum likelihood estimators of the model's parameters. Then, in order to predict a secondary structure, one computes the most likely conformation for the given sequence of bases. The algorithm of Knudsen and Hein [14] is one prominent member of this class using stochastic context-free languages (SCFGs).

When n is the size of the input measured by the number of nucleotides all the algorithms discussed so far have a worst-case running time in $\mathcal{O}(n^3)$ and need space in $\mathcal{O}(n^2)$.

The classical algorithms of Nussinov and Zuker and the SCFG-based algorithms are founded on similar dynamic programming ideas. For a reinterpretation of the algorithms of Nussinov and Zuker in terms of SCFGs see [3].

Recently much attention has been paid to predicting the RNA secondary structure including pseudoknots. Again, one approach is to minimize the free energy of all possible foldings. However, as proven by Lyngsø and Pedersen [16], predicting RNA secondary structure containing pseudoknots of arbitrary types is NP-complete for a large class of reasonable free-energy functions. Thus one has to consider subclasses as done by Rivas and Eddy in [26]. In this paper the authors generalize the dynamic programming paradigm of Zuker to a class of pseudoknots that are restricted to a quite general recursion scheme. The resulting algorithm needs time in $\mathcal{O}(n^6)$ and space in $\mathcal{O}(n^4)$. Reeder and Giegerich [25] turn to a more restricted model for pseudoknots, the so-called canonical simple recursive pseudoknots, which allow for an algorithm with worst case running time in $\mathcal{O}(n^4)$ and space consumption in $\mathcal{O}(n^2)$. Again, the structure is predicted by computing the energetically optimal conformation. However there is no obvious way to restrict the set of pseudo-knot configurations, such that it still covers all biologically relevant structures and is still compatible to efficiently computable context-free grammars.

There are also stochastic models which extend the class of stochastic grammars in use to take care of pseudoknots. One step towards this direction has been taken by Cai et al. [2] who use parallel communicating grammar systems to process pseudoknots leading to an algorithm with running time in $\mathcal{O}(n^6)$ and space in $\mathcal{O}(n^4)$; Uemura et al. [28] changed to tree adjoining grammars yielding also a restricted class of pseudoknots.

The known RNA structures include at most a few pseudo-knots. These can be seen as exceptions, which are added to structures, which follow context-free grammars not allowing pseudo-knots. This point of view is the basis of the approach presented here; a context-free grammar is used to generate the pseudoknot-free core of a structure. Additionally, a special symbol is generated in order to represent regions of the primary structure that will become parts of a pseudoknot.

2 The Model

Like several other groups (as e.g. Knudsen and Hein [14]; Rivas and Eddy [26]; Cai et al. [2]) we use a stochastic grammar to define a prior probability distribution on the set of possible secondary structures. Our grammar (together with a non-standard notion of derivations) allows the generation of pseudoknots in a non-context-free way. Generating an RNA sequence by applying grammar rules obviously does not correspond to a biological mechanism. However, in our Bayesian framework we rather consider it as a way to define a prior distribution on the set of possible RNA structures. We do not intend to put too much information into this prior and therefore use a very simple grammar.

We combine a stochastic context-free grammar (used to generate a pseudoknot-free core structure) with the possibility to add arbitrary stems (in order to insert pseudoknots) in a similar way as described by Cai et al. [2]. The context-free part of our grammar is similar to the grammar of Knudsen and Hein [14]; the non-terminal F generates stems, the non-terminal L generates loops, and the non-terminal S is the starting-symbol of the grammar. In detail, the grammar rules are of the following types,

$$S \to \begin{array}{l} \nearrow\ LS \\ \\ \searrow\ L \end{array} \qquad F \to \begin{array}{l} \nearrow\ xLSy \\ \\ \searrow\ xFy \end{array} \qquad L \to \begin{array}{l} \nearrow\ z \\ \\ \searrow\ uxFyv \\ \\ \searrow\ q \end{array}$$

whereas the terminals $u, v, x, y,$ and z represent nucleotides, i.e. are taken from the set $\mathcal{A} = \{a, c, g, u\}$, and the probabilities of the grammar rules favor pairs (u, v) and (x, y) which match in RNA stems. The terminal symbol q is a surrogate for some subsequence (of the primary structure) which will be part of a non-context-free stem.

The pairs (u, v) and (x, y) in the rule $L \to uxFyv$ are chosen independently of each other according to some distribution $(\pi_{xy})_{xy}$. As usual for SCFGs all rules to be applied to nonterminals are chosen independently of each other. When $L \to z$ is applied, z is taken from a probability distribution $(\pi_a, \pi_c, \pi_g, \pi_u)$. When one of the rules emits a pair of nucleotides, then it is taken from a binucleotide distribution $(\pi_{xy})_{(x,y)\in\{a,c,g,u\}^2}$.

The probabilistic generation of a sequence s of terminals (including q) can be represented by a *parse tree*, which is an ordered tree whose leaves, read from left to right, are labeled by the terminals in sequence s and internal nodes are labeled by nonterminals. The parse tree is generated in the following way: Start with a single node labeled by S. Then repeat the following

three steps until all leaves are labeled by terminals: For each leaf k labeled by some nonterminal X, select some rule $X \to \alpha$, where α is a series of terminals and non-terminals $\alpha_1 \dots \alpha_{n_k}$, add n_k new nodes labeled by $\alpha_1, \dots, \alpha_{n_k}$ and define the node labeled by α_i to be the i-th son of k, $1 \leq i \leq n_k$.

From the perspective of the context-free grammar, the symbol q is a terminal. However, after all nonterminals have been transformed into terminals, the q-symbols are used to generate stems that may belong to pseudoknots. Therefore we need to modify the usual notion of derivation of a context-free grammar: Given a terminal string s, the set of q symbols appearing in s is randomly subdivided into pairs. If the number of q symbols is odd, then one of them is randomly picked and replaced by a nucleotide (single terminal symbol). Each pair of q symbols produces a random number of base pairs. This is done in the following way: Start with $q \to uxRyv$ and iteratetively apply one of the rules $R \to xRy$ or $R \to xy$, where the probability of the rule $R \to xRy$ is the same as the probability of $F \to xFy$ and the emission probabilities for the nucleotide pairs (u, v) and (x, y) are the same as above. Then the two q-symbols within s are replaced by the two complementary halves of the resulting nucleotide sequence. When all q's are replaced, the resulting terminal string s' is considered as the result of the modified derivation process. Note that pseudoknots can be nested in arbitrary ways in this model and that a primary structure in general can be generated in many ways since both, our context-free grammar and the way the symbols q are processed, are ambiguous.

A restriction in our model is that stems cannot be shorter than three base pairs. Since we do not distinguish between bulges and loops, this implies that bulges can only occur in a stem with a distance of at least 3 base pairs. However, the way we combine the SCFG with pseudoknots and the methods described in the following can also be adapted to any other SCFG model of RNA folding without such restrictions.

3 Algorithms and Implementation

McQFold, a C++ implementation of our method is freely available from the web sites of the authors under the terms of the GPL, cf. [8].

3.1 Notations

When a sequence is generated according to our model, we define the *complete history* $\mathcal{H}$ to consist of the parse tree $T(\mathcal{H})$ and the *q-stem configuration* $Q(\mathcal{H})$, which contains the information which q-symbols are paired and which subsequences were emitted from them. If τ is a candidate for the parse tree and ρ a possible *q-stem configuration*, then the probability of the complete history (τ, ρ) is $\Pr(\mathcal{H} = (\tau, \rho)) = \Pr(T(\mathcal{H}) = \tau) \cdot \Pr_{q_\mathcal{H}}(Q(\mathcal{H}) = \rho)$, where $q_\mathcal{H}$ is the number of leaves in τ labeled by q and $\Pr_{q_\mathcal{H}}(Q(\mathcal{H}) = \rho)$ is conditioned on this number. $\Pr(T(\mathcal{H}) = \tau)$ is the product of the probabilities of all rules used to generate τ yielding one factor for each internal node of τ. $\Pr_n(Q(\mathcal{H}) = \rho)$ is 0 if n is not the number of q-symbols occurring in ρ. Otherwise it is the quotient of the probability of the derivation used in order to generate stems from the q-symbols (which again is given by the product of the probabilities of

the rules applied) and the number of possibilities for building pairs of q-regions within τ. This number is $(n-1)(n-3)\cdots 5\cdot 3$ if n is even and $n\cdot(n-2)\cdot(n-4)\cdots 5\cdot 3$ if n is odd. Let $S(h)$ denote the sequence generated by some complete history h. The probability of a sequence s of nucleotides to be equal to $S(\mathcal{H})$ is the sum of the probabilities of all possible histories:

$$\Pr(S(\mathcal{H})=s)=\sum_{\{h\,:\,S(h)=s\}}\Pr(\mathcal{H}=h).$$

Let $J(\mathcal{H})$ be the sequence of intervals $[i_1,j_1],\ldots,[i_p,j_p]$, which indicate the subsequences (given by the i_k-th up to the j_k-th symbol, $1\le k\le p$) of $S(\mathcal{H})$ generated by q-symbols. Let $C(\mathcal{H})$ be the set of unordered pairs $\{a,b\}\subset\{1,\ldots,p\}$ indicating that the subsequences of $S(\mathcal{H})$ given by $J_a(\mathcal{H})=[i_a,j_a]$ and $J_b(\mathcal{H})=[i_b,j_b]$ were jointly generated from a pair of q-symbols.

By $M(\mathcal{H})$ we denote the set of all unordered pairs $\{i,j\}$ such that positions i and j in $S(\mathcal{H})$ are paired in a stem (generated by the context-free part or by q-symbol rules). We call $M(\mathcal{H})$ *the structure* generated by $\mathcal{H}$. Given $M(\mathcal{H})$, the conditional distribution of $S(\mathcal{H})$ is very simple. Each pair $(S_i(\mathcal{H}),S_j(\mathcal{H}))$ with $\{i,j\}\in M(\mathcal{H})$ is distributed according to $(\pi_{xy})_{xy}$, and each $S_i(\mathcal{H})$ with $i\notin\cup_{U\in M(\mathcal{H})}U$ is independently distributed according to $(\pi_x)_x$. There are no further dependencies.

In the sequel, the sequence over $\{a,c,g,u,q\}$ generated by the context-free grammar will only be denoted by $S^q(\mathcal{H})$. Furthermore, $S(\mathcal{H}),M(\mathcal{H}),J(\mathcal{H}),\ldots$ will be denoted by $S,M,J,\ldots$.

3.2 Computing the Likelihood

Given some data (which in our case will be a terminal string s) that are assumed to be randomly generated according to some probability model, the Likelihood of some fixed values of the model's parameters is given by the probability of the data in the model using these parameter settings. In principle, the likelihood for our model can be computed by summing over all possible q-stem-configurations (γ,ι):

$$\begin{aligned}\Pr(S=s) &= \sum_{\gamma,\iota}\Pr(S=s,C=\gamma,J=\iota)\\ &= \sum_{\gamma,\iota}\Pr(S=s\mid C=\gamma,J=\iota)\cdot\Pr(C=\gamma,J=\iota).\end{aligned}$$

For given (γ,ι), the probability $\Pr(S=s\mid C=\gamma,J=\iota)\cdot\Pr(C=\gamma,J=\iota)$ of the sequence s can be computed by dynamic programming strategies known for SCFGs. Instead of computing the sum directly, which is not feasible because of the huge number of possible (γ,ι), one can apply well-known sampling strategies.

3.3 Dynamic Programming Algorithms for SCFGs

For a given sequence $s\in\{a,c,g,u\}^n$ and a set ι of intervals $[i,j]\subset\{1,\ldots,n\}$ let s^ι be the sequence over $\{a,c,g,u,q\}$ that we obtain from s by replacing each subsequence $s_a,\ldots,s_b$

with $[a,b] \in \iota$ by the symbol q. For $1 \leq i \leq j \leq |s^\iota|$ and $A \in \{S, F, L\}$ let $\Phi_{ij}(A)$ be the probability that the context-free part of our grammar transfers the symbol A into the sequence of terminals $s_i^\iota, \ldots, s_j^\iota$. These values can be computed in time $O(|s^\iota|^3)$ by the forward algorithm, which is based on dynamic programming. In detail, the values $\Phi_{ij}(.)$ are computed iteratively with increasing distance $j - i$ and the values $\Phi_{i'j'}(.)$ with $j' - i' < j - i$, which were computed previously, are used for the computation of $\Phi_{ij}(.)$. For example, for the computation of $\Phi_{ij}(F)$ we translate the rules $F \to xFy$ and $F \to xLSy$ into the following equation:

$$\begin{aligned}\Phi_{ij}(F) &= \Phi_{i+1,j-1}(F) \cdot \Pr(F \to xFy) \cdot \pi_{s_i s_j} \\ &\quad + \sum_k \pi_{s_i s_j} \cdot \Pr(F \to xLSy) \cdot \Phi_{i+1,k}(L) \cdot \Phi_{k+1,j}(S).\end{aligned}$$

For computing all these values we need to compute a summand for each of the $O(n^3)$ triples i, k, j with $1 \leq i \leq k \leq j \leq |s^\iota|$. This complexity is caused by the rules $S \to LS$ and $F \to xLSy$. In our implementation we save a significant amount of runtime by keeping track of $N_{ij} = \{k < j \ : \ \Phi_{ik}(L) > 0\}$ and computing only summands with $k \in N_{ij}$. At this point we take advantage of the restriction that stems must contain at least three position pairs and that only mismatches of the type $g : u$ are allowed.

Let $\Psi_{ij}(A)$ be the probability that the parse tree $T(\mathcal{H})$ generates a sequence S^q which begins with the prefix $s_1^\iota, s_2^\iota, \ldots, s_{i-1}^\iota$, is continued by a subsequence generated by a nonterminal A and ends with the suffix $s_{j+1}^\iota, \ldots, s_{|s^\iota|}^\iota$. We compute the values $\Psi_{ij}(.)$ with the outside algorithm (see e.g. [3]), which is also based on dynamic programming. The values $\Phi_{i,j}(.)$ and $\Psi_{ij}(.)$ can be used to compute the posterior probability of i and j to be paired, conditioned on (C, J). For $i, j \notin \cup_{I \in \iota} I$ we obtain

$$\begin{aligned}&\Pr(\{i,j\} \in M \mid C = \gamma, J = \iota, S = s) \\ &= \Psi_{ij}(F) \cdot \Pr(F \to xFy) \cdot \Phi_{i+1,j-1}(F) \cdot \pi_{s_i^\iota, s_j^\iota} \\ &\quad + \sum_{k=i+1}^{j-2} \Psi_{ij}(F) \cdot \Pr(F \to xLSy) \cdot \Phi_{i+1,k}(L) \cdot \Phi_{k+1,j-1}(S) \cdot \pi_{s_i^\iota, s_j^\iota} \\ &\quad + \Psi_{ij}(L) \cdot \Pr(L \to uxFyv) \cdot \Phi_{i+2,j-2}(F) \cdot \pi_{s_i^\iota, s_j^\iota} \cdot \pi_{s_{i+1}^\iota, s_{j-1}^\iota} \\ &\quad + \Psi_{i-1,j+1}(L) \cdot \Pr(L \to uxFyv) \cdot \Phi_{i+1,j-1}(F) \cdot \pi_{s_{i-1}^\iota, s_{j+1}^\iota} \cdot \pi_{s_i^\iota, s_j^\iota}.\end{aligned}$$

3.4 Bayesian Sampling

Given an RNA sequence s we want to sample sets of position pairs μ according to their posterior probability of μ, which is given by

$$\begin{aligned}\Pr(M = \mu \mid S = s) &= \sum_{\gamma,\iota} \Pr(M = \mu, C = \gamma, J = \iota \mid S = s) \\ &= \sum_{\gamma,\iota} \Pr(M = \mu \mid C = \gamma, J = \iota, S = s) \cdot \Pr(C = \gamma, J = \iota \mid S = s).\end{aligned}$$

We sample values (γ, ι) for (C, I) according to the posterior probability $\Pr(C = \gamma, J = \iota \mid S = s)$. For given (γ, ι, s) we compute for all position pairs $\{i, j\}$ the probability to be in M. This is done with a variant of the inside-outside algorithm in time $O(n^3)$ and space $O(n^2)$. We approximate $\Pr(\{i, j\} \in M \mid S = s)$ by averaging $\Pr(\{i, j\} \in M \mid C = \gamma, J = \iota, S = s)$ over all sampled pairs (γ, ι).

In a similar fashion we sample possible RNA structures μ (approximately) from their posterior probability $\Pr(M = \mu \mid S = s)$. For each sampled pair (γ, ι) we apply the inside algorithm with randomized backtracking for sampling possible RNA structures μ according to $\Pr(M = \mu \mid C = \gamma, J = \iota, S = s)$.

3.4.1 Metropolis-Hastings-Strategy for Sampling q-Stem Configurations

We apply a Markov-chain Monte-Carlo method (cf.[15]) for sampling candidates (γ, ι) for (C, I) according to the posterior probability $\Pr(C = \gamma, J = \iota \mid S = s)$. We construct a random sequence $(C_0, J_0), (C_1, J_1), \ldots$ such that $\Pr((C_n, J_n) = (\gamma, \iota))$ converges to $\Pr(C = \gamma, J = \iota \mid S = s)$ for $n \to \infty$. This is done with a Metropolis-Hastings strategy (cf. [17][9][15]). Given $(C_{k-1}, J_{k-1}) = (\gamma, \iota)$ we let a random generator propose $(C_k, J_k) = (\gamma', \iota')$ with some probability $Q_{(\gamma,\iota)\to(\gamma',\iota')}$ and accept this proposal with probability

$$\min\left\{1\ ,\ \frac{Q_{(\gamma',\iota')\to(\gamma,\iota)}}{Q_{(\gamma,\iota)\to(\gamma',\iota')}} \cdot \frac{\Pr(C = \gamma', J = \iota' \mid S = s)}{\Pr(C = \gamma, J = \iota \mid S = s)}\right\}.$$

It is well-known and easy to check that this results in a Markov chain which converges to the desired probability distribution if the "proposal chain" $Q_{\cdot\to\cdot}$ is irreducible.

In the examples described in section 4 we start with $J = \emptyset$ and perform a "burn-in" of 100 steps before we start sampling. Then we sample 20 q-stem configurations, always performing 50 Metropolis-Hastings steps between two subsequent samplings.

3.4.2 Proposal Distribution

In order to obtain a proposal (γ', ι') for (C_k, J_k) from the value (γ, ι) of (C_{k-1}, J_{k-1}) , we toss a fair coin to decide if we erase (if possible) or add a pair of intervals, which form a q-stem. If $|\iota| = 0$, we will always propose to add one stem.

If we decide to erase a pair of q-regions, then we pick $\{[i_a, j_a], [i_b, j_b]\}$ uniformly from ι and set $\iota' := \iota \setminus \{[i_a, j_a], [i_b, j_b]\}$ and $\gamma' := \gamma \setminus \{\{[i_a, j_a], [i_b, j_b]\}\}$.

If we decide to add one q-stem, then we draw a pair of intervals $\xi = \{[i_x, j_x], [i_y, j_y]\}$ from a certain probability distribution $(p_\xi)_\xi$. The probability p_ξ reflects how suspicious ξ is to be part of a pseudoknot. This distribution is computed in advance. p_ξ is supposed to be high if s's subsequences corresponding to $[i_x, j_x]$ and $[i_y, j_y]$ fit together well, i.e. represent reverse-complementary subsequences at best, and whether it is unlikely that ξ is part of a regular stem which can be produced by the context-free part of the grammar.

In order to quantify the first criterion we reverse the sequence s, replace all nucleotides by their complements, and align the resulting sequence locally against the original sequence. An ungapped local alignment is called HSP (high-scoring segment-pair, cf. [1][22][19]) if its score

is higher than some threshold and if the alignment is not part of an ungapped alignment of even higher score. We use $\log \frac{\pi_{uv}}{\pi_u \cdot \pi_v}$ as alignment score for a pair of nucleotides u and v. We select our q-stem proposals among all pairs of intervals which correspond to HSPs. Let σ_ξ be the score of the HSP corresponding to ξ.

For each HSP we test the second criterion by SCFG methods. Using the inside-outside algorithm we compute for each pair of positions i and j the probability $\Pr(\{i,j\} \in M \mid S = s, C = J = \emptyset)$ that i and j are paired in a structure without pseudoknots. Let q_ξ be the maximum of these values over all pairs $\{i,j\}$ implied by ξ. We use

$$p_\xi \propto \frac{1 - e^{-\sigma_\xi \cdot c_1}}{\max\{q_\xi \ , \ c_2\}}$$

with $c_1 = 10^{-6}$ and $c_2 = 10^{-5}$. This function is the result of manual trial-and-error optimization with simulated sequences of length ~ 300. We expect potential for improvements of our method at this point, especially for longer sequences.

3.5 Searching the Most Probable Folding

For estimating the structure of an RNA molecule from its sequence we strongly recommend using Bayesian sampling instead of searching a single structure with maximal probability or energy. However, for being able to compare our model to other models of RNA folding we have also implemented a heuristic optimization method, which is based on simulated annealing (cf. [13][15]). In the first stage it searches for the q-stem configuration (γ, ι) with the highest posterior probability $\Pr((C,J) = (\gamma,\iota) \mid S = s)$. For this end, we start with the best (γ_0, ι_0) found during the MCMC part. In the following 1000 steps we take proposals (γ', ι') to change $(\gamma_{i-1}, \iota_{i-1})$ into (γ_i, ι_i) from the same proposal chain as in the MCMC procedure (see section 3.4.2), but the acceptance probability is now given by

$$\min\left\{1, \left(\frac{\Pr((C,J) = (\gamma',\iota') \mid S = s)}{\Pr((C,J) = (\gamma_{i-1},\iota_{i-1}) \mid S = s)}\right)^{1/t_i}\right\},$$

setting the so-called temperature t_i in the i-th step to $i/300$.

In the second stage we start with the optimum found so far and perform additional 1000 steps of the same proposal chain using the same temperatures $t_1 = 300, t_2 = 150, t_3 = 100, \ldots, t_{1000} = 3/10$. But now the acceptance probability in step i is taken to be

$$\min\left\{1, \left(\frac{\max_\tau \Pr(T = \tau, (C,J) = (\gamma',\iota') \mid S = s)}{\max_\tau \Pr(T = \tau, (C,J) = (\gamma_{i-1},\iota_{i-1}) \mid S = s)}\right)^{1/t_i}\right\}.$$

There, for given γ, ι, s, the value $\max_\tau \Pr(T = \tau, (C,J) = (\gamma,\iota) \mid S = s)$ can efficiently be computed with the SCFG variant of the CYK algorithm, which is similar to the forward algorithm with sums replaced by maximizations (cf. [3]).

The idea behind this two-stage approach is as follows. In the first stage we look for values for (C,J), which contribute much probability in combination with several different choices for

T. In the second stage, when we are closer to the optimum, we look for the most probable triple of values $\widehat{(\gamma, \iota, \tau)}$ for (C, J, T). The most probable triple found during the second stage is an estimate h for the complete history $\mathcal{H}$ of the sequence s. The corresponding RNA structure $M(h)$ is the output of our program McQFold.

Of course, it is not guaranteed that our simulated-annealing heuristic finds the most probable history. Moreover, since the map $h \mapsto M(h)$ from the possible histories to the possible structures of a sequence is not injective, the most probable history does not necessarily correspond to the most probable structure. Note that there is no such problem when Bayesian sampling is applied. If a random history H is sampled from the posterior distribution, then $M(H)$ is distributed according to the posterior distribution on RNA structures.

3.6 What is Possible in Our Model but Ignored by Our Method

According to our model, an odd number of q-symbols can be generated. In this case, one of them will be replaced by a nucleotide. The contribution of this possibility to posterior probabilities of structures is rather small and we neglect it in our algorithms. Because we change the configuration of q-stems only during the proposal procedures by adding or removing pairs of q-symbols which jointly generate a q-stem, no odd number of q-symbols may occur. Note that, assuming reasonable values of the model parameters, a history containing the rule $q \to x$ (for $x \in \{a, c, g, u\}$) can never be the most probable history of a given sequence, because we obtain a more probable history by replacing the sequence of rules $L \to q \to x$ by the single, more probable rule $L \to x$.

When looking for q-stem candidates we always extend the high-scoring segment pairs as far as possible. As a consequence, we cannot exactly detect a q-stem if its neighboring positions match by chance such that the stem can be elongated to a stem of higher score.

4 Results

In the examples below our program McQFold is used with the following probabilities for the grammar rules: $S \to LS : 0.85,\ S \to L : 0.15,\ F \to xLSy : 0.25,\ F \to xFy : 0.75,\ L \to x : 0.9,\ L \to uxFyv : 0.06,\ L \to q : 0.04$. These values were not specifically fitted to sequence data. We simulated data according to our model and changed the parameters until the numbers and lengths of loops, stems and pseudoknots seemed reasonable to us. Again, we expect potential for improvements.

For the nucleotide distribution in loops and the base pair probabilities we used the following rough estimates from the available sequence data: $\pi_{au} = \pi_{ua} = 0.175,\ \pi_{cg} = \pi_{gc} = 0.275,\ \pi_{gu} = \pi_{ug} = 0.05,\ \pi_a = 0.35,\ \pi_c = 0.2,\ \pi_g = 0.2,\ \pi_u = 0.25$.

We do not allow for other mismatches than $g : u$ in stems, or the other way round, other mismatches are interpreted as bulges.

Results of McQFold are compared to those of RNAfold from the Vienna package (cf. [11][10]), and of pknotsRG, an implementation of Reeder and Giegerich's method, cf. [25]. All programs

	correct pairs	false pairs	missing pairs	specificity	sensitivity
McQFold	86	21	21	80.4 %	80.4 %
RNAfold	58	61	49	48.7 %	54.2 %
pknotsRG	55	65	52	45.8 %	51.4 %

Table 1: Ability of McQFold, RNAfold, and pknotsRG to correctly predict the base pairs in the tmRNA of Treponema Pallidum. As reference we take the structure given on the tmRNA website [30].

were used with default parameter settings. We judge the results of the three methods by the following two criteria:

Specificity: The relative frequency of correctly predicted pairs among all position pairs that were predicted to be paired with each other.

Sensitivity: The relative frequency of correctly predicted pairs among all position pairs that are actually paired in a stem.

4.1 Illustrative Example: Folding pre-tmRNA of Treponema Pallidum

As an illustrative example we consider a pre-tmRNA sequence found in the syphilis bacterium treponema pallidum (tmRNA Website ID: Trepo_palli_0191, cf. [6][31][30]).

The posterior probabilities for base pairings estimated by McQFold are shown in the upper half of Figure 1(a). The lower half displays the structure given on the tmRNA website [30], which we assume to be true. Most of the actual stems were predicted as highly probable and vice versa. In Figures 1(b),(c),(d) we compare the estimated posterior probabilities to the structure given on the tmRNA website. In this example the structure obtained by the simulated annealing procedure of McQFold was slightly more accurate than the very similar structures given by RNAfold and pknotsRG.

Detecting pseudoknots in this sequence was obviously a difficult task. Only McQFold was able to correctly predict one of the pseudoknots.

4.2 Performance on tmRNA

Additionally, we applied McQFold, RNAfold, and pknotsRG to all 351 tmRNA molecules of length > 200 found on the tmRNA website (http://www.indiana.edu/$\sim$tmrna/ , cf. [31][30]) in autumn 2005. In Figure 2 we compare the distributions of the specificity and the sensitivity reached by our method and by the programs RNAfold and pknotsRG over these 351 sequences. Our method is slightly, but significantly better in specificity than the other methods. The median sensitivity is similar for the three methods. Our method has the least variability in this value.

The performance of our method on tmRNA sequences is very encouraging. However, it is not yet clear if this indicates a great potential of our model or if the parameters we used with our

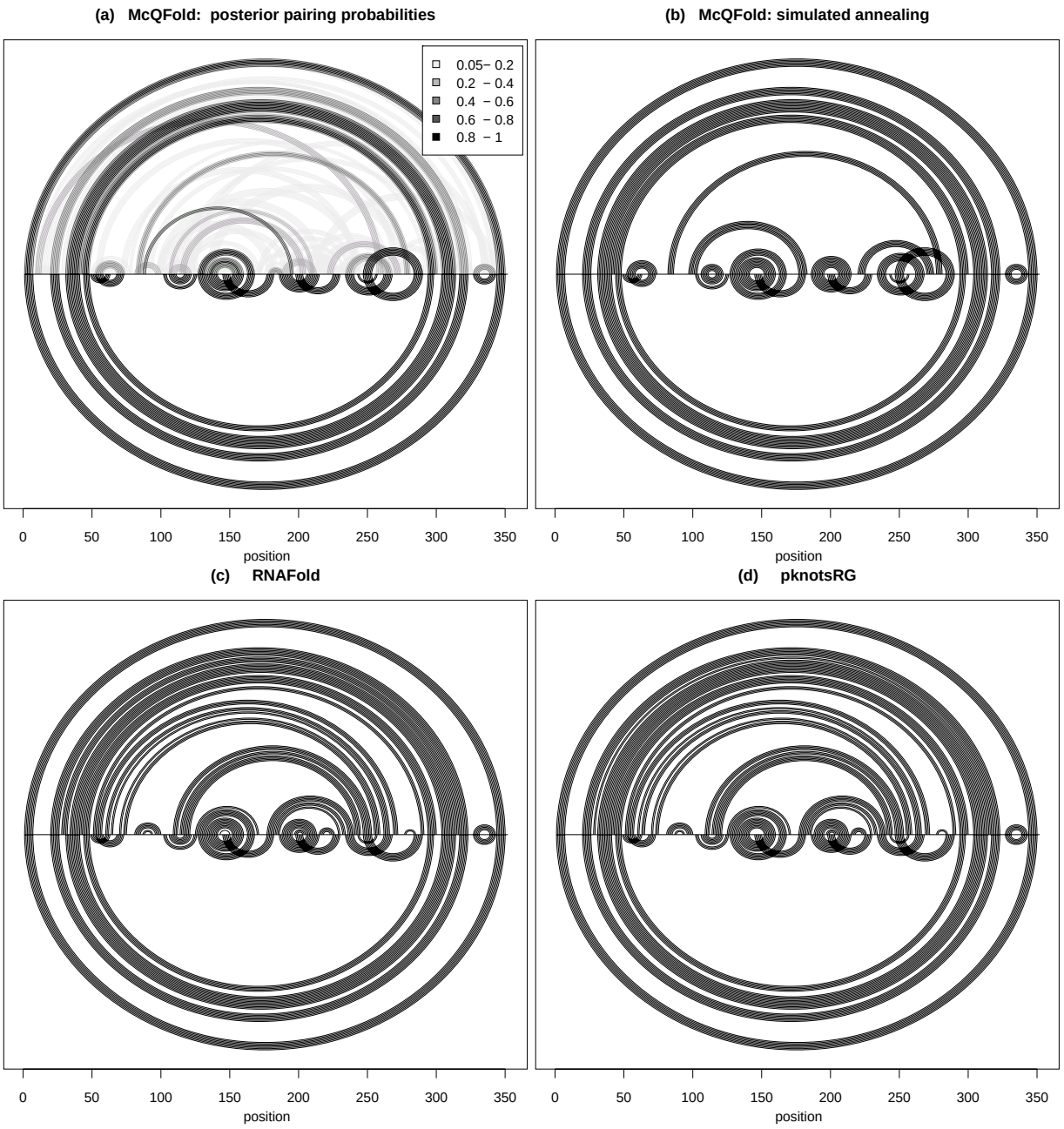

Figure 1: Results of McQFold for the pre-tmRNA of Treponema Pallidum. Paired nucleotides are represented by arcs connecting the positions. The Lower half of each sub-figure displays the structure given on the tmRNA website Upper halves: (a) grey tones code for the posterior pairing probabilities estimated by the McQFold's MCMC procedure, (b) structure estimated by the McQFold's simulated annealing procedure, (c) structure estimated by RNAfold, cf. [10], (d) structure estimated by pknotsRG, cf. [25].

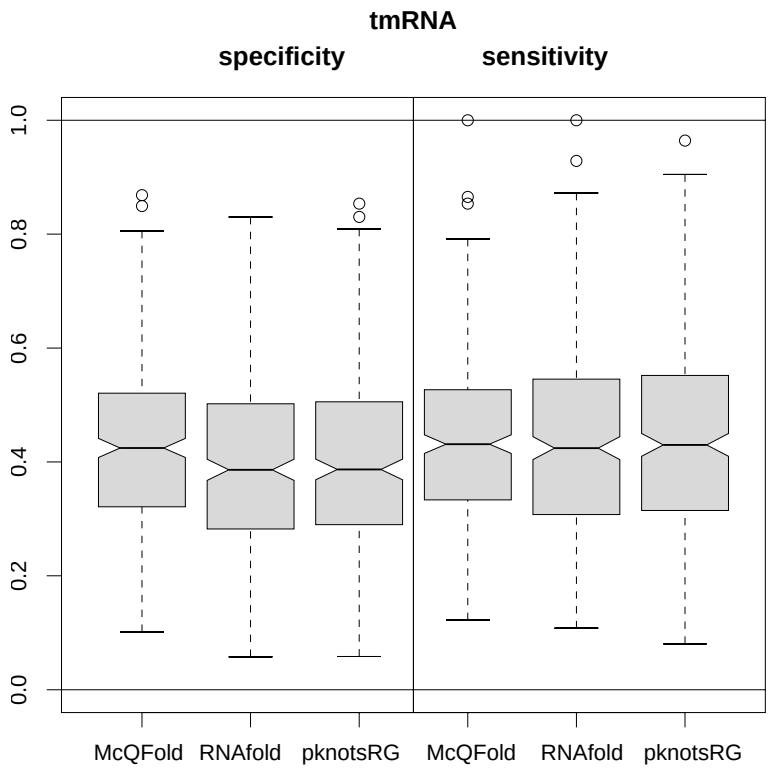

Figure 2: Comparison of sensitivity and specificity distributions of McQFold, RNAfold, and pknotsRG on tmRNA sequences. All programs were used with default parameter settings.

method were just by chance more suitable specifically for tmRNA than the default parameters we used with RNAfold and pknotsRG.

Figure 2 also makes clear that we cannot expect to find the true secondary structure of a tmRNA molecule by applying any of the three programs to its sequence. This uncertainty should be made explicit and it is one of our reasons to favor Bayesian sampling over point estimation for RNA folding. In Figure 3 we explore the reliability of our Bayesian sampling method (cf. section 3.4.1). For $0 \leq x \leq 1$ we consider all pairs of positions to which a pairing probability of $\approx x$ was assigned. If our method works well, then among all these pairs the ratio of position pairs that are actually paired in the true RNA structure should be close to x. Indeed, Figure 3 shows that this is the case.

4.3 Performance on Simulated Data

Could we improve the method by using a more realistic model or better model parameters? In order to get an impression of the possible performance of our method under ideal conditions, we simulate data according to the model our algorithms are based on, using the same model parameter values for data simulation as for data analysis in McQFold (at least up to the neglects discussed in section 3.6). Sequences of length < 300 were rejected. Thus, we generated 200 folded sequences of length between 300 and 460 and applied McQFold, RNAfold and pknotsRG

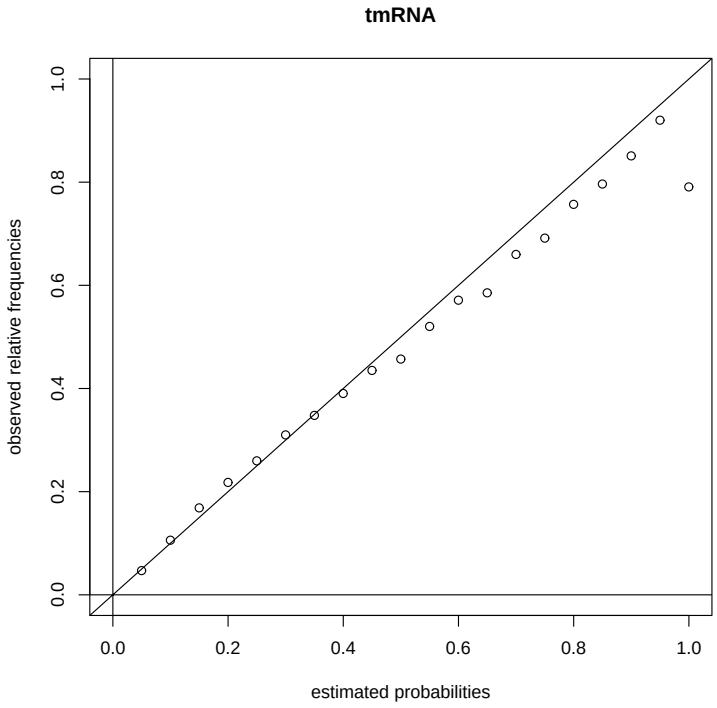

Figure 3: Each dot (x, y) displays the ratio y of actually paired position pairs among all position pairs in any of the tmRNA sequences to which McQFold assigned a folding probability of $\approx x$.

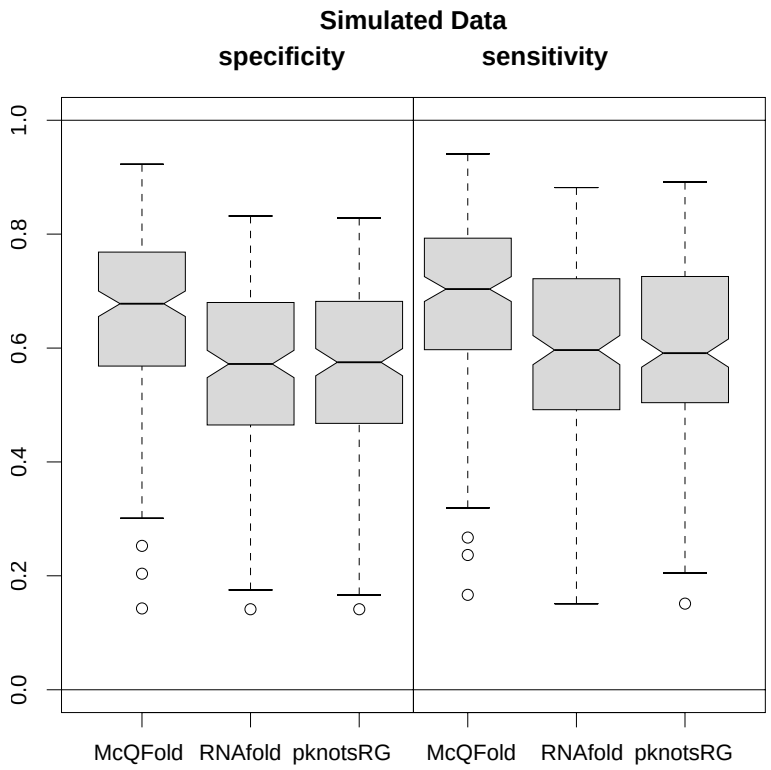

Figure 4: Comparison of McQFold, RNAfold, and pknotsRG when applied to random data generated according to our probability model.

to these sequences and compared the predicted to the actual structures. It is not astonishing that McQFold performed better than the other programs for these sequences, cf. Figure 4. However, for most sequences only between 50% and 85% of the position pairs were correctly predicted, and the variance of this value differs very much between the sequences.

Figure 5 shows that this variance can be assessed quite well by our MCMC sampling method. For all values of x between 0 and 1 we see that among all position pairs with estimated posterior pairing probability around x, the relative frequency of correct position pairs was indeed close to x.

5 Discussion

We have presented a model for RNA structures with pseudoknots and Bayesian sampling methods for estimating the structure of RNA molecules including pseudoknots. We do not restrict the entanglement of pseudoknots a priori but our randomized procedures may take advantage from the observation that structures containing a very high number of pseudoknots are unlikely for realistic RNA sequences.

The examples in section 4 illustrate that estimating the secondary structure of an RNA molecule just from its nucleotide sequence is a difficult task. The reliability of our method

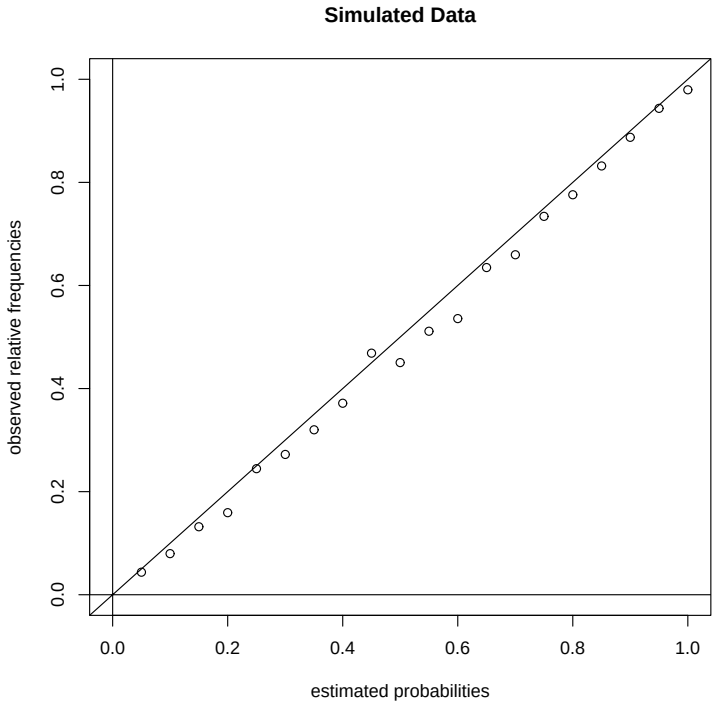

Figure 5: Each dot (x, y) displays the ratio y of actually paired position pairs among all position pairs in any of the simulated sequences to which McQFold assigned a folding probability of $\approx x$.

(even without careful parameter-adjustments) for this problem is similar to that of RNAFold and pknotsRG. None of the methods/programs came close to the presumably real structures which are given on the tmRNA website. If additional information (as for example known structures of similar sequences) is not available, it is important to assess the reliability of the structure estimates. For our model this can be done with the MCMC-sampling strategy described in section 3.4.1. In fact, we obtained very encouraging results in section 4 when we applied this Bayesian method to sets of test data. For the tmRNA sequences and the simulated data the estimates of the reliability of position pairings turned out to be very accurate.

If more information than just one sequence is available, it should obviously be used. In many cases homologous RNA sequences with known structure can be found in a data base. Homology between sequences can help even if there is no prior knowledge about the structure of any of the sequences. Compensatory mutations, which conserve complementarity between segments of an RNA sequence through an alignment give strong evidence that the segments form a stem which is relevant for the function of the molecule, cf. [12]. Extracting structural information like this from an alignment of RNA sequences should also be based on a probabilistic model for RNA structure. In this article we have shown how pseudoknots can be incorporated in such a model. If it seems necessary, it is straight-forward to combine the pseudoknot generating part of the grammar with a more elaborate SCFG, as for example the SCFG proposed by Nebel in [23].

Finally, information on the structure of related RNA molecules might be useful for refining the alignment of their sequences. This combined with the desire to assess the reliability of the estimates for alignments and structures leads to the idea of applying a Gibbs-sampling strategy (cf. [7][15]) to combine our RNA structure estimation method with sampling algorithms for sequence alignments (cf. [20][18][5][21]).

References

[1] S.F. Altschul, R. Bundschuh, R. Olsen, T. Hwa. The estimation of statistical parameters for local alignment score distributions. Nucleic Acids Research 29:351-361, 2001.

[2] L. Cai, R. L. Malmberg and Y. Wu. Stochastic modeling of RNA pseudoknotted structures: a grammatical approach. Bioinformatics 19: i66-i73, 2003.

[3] R.L. Durbin, S.R. Eddy, A. Krogh, G. Mitchison. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge, UK, 1998.

[4] C. Färber. Sampling-Algorithmen für stochastische Grammatiken zur Vorhersage von RNA-Sekundärstrukturen mit Pseudoknoten. Diploma Thesis, Institut für Informatik, J.W. Goethe-Universität Frankfurt am Main, Germany, 2005.

[5] R. Fleißner, D. Metzler, A. von Haeseler. Simultaneous Statistical Multiple Alignment and Phylogeny Reconstruction. Syst. Biol. 54:548-561, 2005.

[6] C. M. Fraser, S.J. Norris, G.M. Weinstock, O. White, G.G. Sutton, R. Dodson, M. Gwinn, E.K. Hickey, R. Clayton, K.A. Ketchum, E. Sodergren, J.M. Hardham, M.P. McLeod, S. Salzberg, J. Peterson, H. Khalak, D. Richardson, J.K. Howell, M. Chidambaram, T. Utterback, L. McDonald, P. Artiach, C. Bowman, M.D. Cotton, C. Fujii, S. Garland, B. Hatch, K. Horst, K. Roberts, M. Sandusky, J. Weidman, H.O. Smith, J.C. Venter. Complete genome sequence of Treponema pallidum, the syphilis spirochete. Science 281(5375):324-5, 1998.

[7] S. Geman, D. Geman. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. IEEE Transaction on Pattern Analysis and Machine Intelligence, 6:721-741, 1984

[8] GPL. The GNU Public License. Available in full from http://www.fsf.org/copyleft/gpl.html 2000.

[9] W.K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika 57:97-109, 1970.

[10] I. L. Hofacker. Vienna RNA secondary structure server. Nucleic Acids Research 31(13):3429-3431, 2003.

[11] I.L. Hofacker, W. Fontana, P.F. Stadler, L. S. Bonhoeffer, M. Tacker, P. Schuster. Fast Folding and Comparison of RNA Secondary Structures. Monatsh. Chem. 125:167-188, 1994.

[12] M. Kimura. The role of compensatory neutral mutations in molecular evolution. J. Genet. 64:7-19, 1985.

[13] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi. Optimization by simulated annealing. Science 220:671-680, 1983.

[14] B. Knudsen and J. Hein. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. Bioinformatics 15: 446-454, 1999.

[15] J.S. Liu. Monte Carlo Strategies in Scientific Computing. Springer, 2001.

[16] R.B. Lyngsø, C.N. Pedersen. RNA Pseudoknot Prediction in Energy-Based Models. J. Comput Biol. 7(3):409-427, 2000.

[17] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller. Equations of state calculations by fast computing machines. Journal of Chemical Physics 21(6):1087-1092, 1953.

[18] D. Metzler. Statistical Alignment based on fragment insertion and deletion models. Bioinformatics 19(4):490-499, 2003

[19] D. Metzler. Robust E-Values for Gapped Local Alignments. Journal of Computational Biology 13(4):882-896, 2006.

[20] D. Metzler, R. Fleißner, A. Wakolbinger, A. von Haeseler. Assessing variability by joint sampling of alignments and mutation rates. J. Mol. Evol. 53(6):660-669, 2001.

[21] D. Metzler, R. Fleißner, A. Wakolinger, A. v. Haeseler. Stochastic insertion-deletion processes and statistical sequence alignment. In: J.D. Deuschel, A. Greven (Eds.) Interacting Stochastic Systems, Springer, 2005.

[22] D. Metzler, S. Grossmann, A. Wakolbinger. A Poisson model for gapped local alignments. Stat. Prob. Letters 60:91-100, 2002.

[23] M.E. Nebel. Identifying Good Predictions of RNA Secondary Structure. Proceedings of the Pacific Symposium on Biocomputing 2004, 423-434, 2004.

[24] R. Nussinov, G. Pieczenik, J. R. Griggs and D. J. Kleitman. Algorithms for loop matchings. SIAM Journal of Applied Mathematics 35: 68-82, 1978.

[25] J. Reeder and R. Giegerich. Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. BMC Bioinformatics, 5:104, 2004.

[26] E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. J. Mol. Biol. 285:2053-2068, 1999.

[27] I. Tinoco, P. N. Borer, B. Dengler, M. D. Levine, O. C. Uhlenbeck, D. M. Crothers and J. Gralla. Improved estimation of secondary structure in ribonucleic acids. Nature New Biology, 246:40-41, 1973.

[28] Y. Uemura, A. Hasegawa, S. Kobayashi and T. Yokomori. Tree adjoining grammars for RNA structure prediction. Theoretical Computer Science 210:277-303, 1999.

[29] M. S. Waterman. Secondary Structure of Single-Stranded Nucleic Acids, Advances in Mathematics Supplementary Studies 1:167-212, 1978.

[30] K.P. Williams. The tmRNA Website. Nucleic Acids Research, 28(1):168, 2000.

[31] K.P. Williams, D.P. Bartel. The tmRNA Website. Nucleic Acids Research, 26(1):163-165, 1997.

[32] M. Zuker and D. Sankoff. RNA Secondary Structures and Their Prediction, Bulletin of Mathematical Biology 46:591-621, 1984.

[33] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res. 9:133-148, 1981.