A n^2 RNA SECONDARY STRUCTURE PREDICTION ALGORITHM

Markus E. Nebel and Anika Scheid*

Department of Computer Science, University of Kaiserslautern, P.O. Box 3049, D-67653 Kaiserslautern, Germany

Keywords: RNA folding, RNA secondary structure, Computational prediction, Probabilistic modeling, Stochastic context-free grammars, Statistical sampling, Inside-outside calculation, Heuristic approximation.

Abstract: Several state-of-the-art tools for predicting RNA secondary structures have worst-case time and space requirements of $O(n^3)$ and $O(n^2)$ for sequence length n, limiting their applicability for practical purposes. Accordingly, biologists are interested in getting results faster, where a moderate loss of accuracy would willingly be tolerated. For this reason, we propose a novel algorithm for structure prediction that reduces the time complexity by a linear factor to $O(n^2)$, while still being able to produce high quality results. Basically, our method relies on a probabilistic sampling approach based on an appropriate *stochastic context-free grammar (SCFG)*: using a well-known or a newly introduced sampling strategy it generates a random set of candidate structures (from the ensemble of all feasible foldings) according to a "noisy" distribution (obtained by heuristically approximating the inside-outside values) for a given sequence, such that finally a corresponding prediction can be efficiently derived. Sampling can easily be parallelized. Furthermore, it can be done in-place, i.e. only the best (most probable) candidate structure generated so far needs to be stored and finally communicated. Together, this allows to efficiently handle increased sample sizes necessary to achieve competitive prediction accuracy in connection with the noisy distribution.

1 INTRODUCTION

Over the past years, several new approaches towards the prediction of RNA secondary structures from a single sequence have been invented which are based on generating statistically representative and reproducible samples of the entire ensemble of feasible structures for the given sequence. For example, the popular Sfold software (Ding and Lawrence, 2003; Ding et al., 2004) employs a sampling extension of the partition function (PF) approach (McCaskill, 1990) to produce statistically representative subsets of the Boltzmann-weighted ensemble. More recently, a corresponding probabilistic method has been studied (Nebel and Scheid, 2011) which actually samples the possible foldings from a distribution implied by a sophisticated stochastic context-free grammar (SCFG).

Notably, both sampling methods can be extended for predicting secondary structures in $O(n^3)$ time and with $O(n^2)$ space requirements, respectively. These worst-case complexities are actually identical to those of modern state-of-the-art tools for computational structure prediction from a single sequence, for instance the commonly used minimum free energy (MFE) based Mfold (Zuker, 1989; Zuker, 2003) and Vienna RNA (Hofacker et al., 1994; Hofacker, 2003) packages or the popular SCFG based Pfold software (Knudsen and Hein, 1999; Knudsen and Hein, 2003). Furthermore, applications to structure prediction showed that neither of the two competing sampling approaches (SCFG and PF based method) generally outperforms the other and consequently, it is not obvious which one should rather be preferred in practice. This somehow contradicts the fact that the best physics-based prediction methods still generally perform significantly better than the best probabilistic approaches. In principle, only if the computational effort of one particular variant could be improved without significant losses in quality (that is if one of them required considerably less time than the others while it sacrificed only little predictive accuracy), then the corresponding method would be undoubtably the number one choice for practical applications, indeed outperforming all other modern computational tools for predicting the secondary structure of RNA sequences. This, by the way, due to the often quite large sizes of native RNA molecules consid-

*Corresponding author. The research of this author has been supported by the Carl-Zeiss-Stiftung.

E. Nebel M. and Scheid A..
A n2 RNA SECONDARY STRUCTURE PREDICTION ALGORITHM.
DOI: 10.5220/0003764600660075
In *Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms* (BIOINFORMATICS-2012), pages 66-75
ISBN: 978-989-8425-90-4

ered in practice, meets exactly the demands imposed by biologists on computational prediction procedures: rather getting moderately less accurate (but still good quality) results in less time than needing significantly more time for obtaining results that are expectedly not considerably more accurate.

Note that recently, there already have been several practical heuristic speedups (Wexler et al., 2007; Backofen et al., 2011). Particularly, the approach of (Wexler et al., 2007) for folding single RNA sequences manages to speed up the standard dynamic programming algorithms without sacrificing the optimality of the results, yielding an expected time complexity of $O(n^2 \cdot \psi(n))$, where $\psi(n)$ is shown to be constant *on average* under standard polymer folding models; in (Backofen et al., 2011), it is shown how to reduce those average-case time and space complexities in the sparse case. Furthermore, the practical technique from (Frid and Gusfield, 2010) achieves an improved worst-case time complexity of $O(n^3/\log(n))$, and with the (MFE and SCFG based) algorithms from (Akutsu, 1999), a slight worst-case speedup of $O(n^3 \cdot \log(\log(n))^{1/2}/\log(n)^{1/2})$ time can be reached (whose practicality is unlikely and unestablished).

In this article, we present a new way to reduce the worst-case time complexity of SCFG based statistical sampling by a linear factor, making it possible to predict for instance the most probable (MP) structure among all feasible foldings for a given input sequence of length n (in direct analogy to conventional structure prediction via SCFGs) with only $O(n^2)$ time and space requirements. This complexity improvement is basically realized by employing an appropriate heuristic instead of the corresponding exact algorithm for preprocessing the input sequence, i.e. for deriving a "noisy" distribution (induced by heuristic approximations of the corresponding inside and outside probabilities) on the entire structure ensemble for the input sequence. From this distribution candidate structures can be efficiently sampled.[2] Moreover, we will consider two different sampling strategies: (a slight modification of) the widely known sampling procedure from (Ding and Lawrence, 2003; Nebel and Scheid, 2011) which basically generates a random structure from outside to inside, and a novel alternative strategy that obeys to contrary principles and employs a reverse course of action (from inside to outside) but manages to take more advantage of the approximative preprocessing.

As we will see, even building on our new heuristic preprocessing step, both sampling strategies can be applied to obtain MP structure predictions of respectable accuracy. In principle, for sufficiently large sample sizes we obtain a similar high predictive accuracy as in the case of exact calculations[3]. The seemingly sole pitfall is that due to the noisy ensemble distribution resulting from approximative computations, the resulting samples are no longer guaranteed to primarily contain rather likely structures (with respect to the *exact* distribution of feasible foldings for a given input sequence), such that we usually have to generate more candidate structures (i.e., consider larger sample sizes) in oder to ensure reproducible structure predictions. However, this is quite unproblematic in practice: firstly, we can generate the candidate structures in-place (only the so far most probable structure needs to be stored), such that large sample sizes give no rise to memory consumption and secondly, generating samples can easily be parallelized on modern multi-core architectures or grids.

2 PRELIMINARIES

In the sequel, given an RNA molecule r consisting of n nucleotides, we denote the corresponding sequence fragment from position i to position j, $1 \leq i \leq j \leq n$, by $R_{i,j} = r_i r_{i+1} \dots r_{j-1} r_j$. Accordingly, $S_{i,j}$ denotes a feasible secondary structure on $R_{i,j}$.

2.1 Sampling based on SCFG Model

Briefly, probabilistic sampling based on a suitable SCFG $\mathcal{G}_s$ with sets $I_{\mathcal{G}_s}$ and $\mathcal{R}_{\mathcal{G}_s}$ of intermediate symbols and productions, respectively, and axiom $S \in I_{\mathcal{G}_s}$ (that models the class of all feasible secondary structures) has two basic steps: In the first step (preprocessing), all inside probabilities

$$\alpha_X(i,j) := \Pr(X \Rightarrow_{lm}^{*} r_i \dots r_j) \qquad (1)$$

and all outside probabilities

$$\beta_X(i,j) := \Pr(S \Rightarrow_{lm}^{*} r_1 \dots r_{i-1}\ X\ r_{j+1} \dots r_n) \qquad (2)$$

for a sequence r of size n, $X \in I_{\mathcal{G}_s}$ and $1 \leq i, j \leq n$, are computed. According to (Nebel and Scheid, 2011), this can be done with a special variant of an Earley-style parser (such that the considered grammar does not need to be in *Chomsky normal form (CNF)*),

[2]With purposive proof-of-concept implementations (in Wolfram Mathematica 7.0), for instance the overall preprocessing time for *E.coli* tRNA^{Ala} (of length $n = 76$) could be reduced from 49.0 (traditional cubic algorithm) to only 3.7 (new quadratic strategy) seconds.

[3]For *E.coli* tRNA^{Ala}, we for instance observed the same sensitivity and specificity values of 1.0 and 0.91, respectively, with a particular application of our heuristic method and the corresponding exact variant.

where the grammar parameters (trained beforehand on a suitable RNA structure database) are splitted into a set of *transition probabilities* $\mathrm{Pr}^0_{tr}(rule)$ for $rule \in \mathcal{R}_{\mathcal{G}_s}$ and two sets of *emission probabilities* $\mathrm{Pr}^1_{em}(\cdot)$ for the 4 unpaired bases and the 16 possible base pairings. For any such SCFG $\mathcal{G}_s$, there results $O(n^3)$ time complexity and $O(n^2)$ memory requirement for this preprocessing step. Note that in this work, we will use the sophisticated grammar from (Nebel and Scheid, 2011) which has been parameterized to impose two relevant restrictions on the class of all feasible structures, namely a minimum length of $\min_{HL}$ for hairpin loops and a minimum number of $\min_{\mathrm{hel}}$ consecutive base pairs for helices.

The second step takes the form of a recursive sampling algorithm to randomly draw a complete secondary structure by consecutively sampling substructures (defined by base pairs and unpaired bases). Notably, different sampling strategies may be employed for realizing this step; two contrary variants that will be considered within this work are described in detail in Section 4. In general, for any sampling decision (for example choice of a new base pair), the strategy considers the respective set of all possible choices that might actually be formed on the currently considered fragment of the input sequence. Any of these sets contains exactly the mutually and exclusive cases as defined by the alternative productions (of a particular intermediate symbol) of the underlying grammar. The corresponding random choice is then drawn according to the resulting conditional sampling distribution (for the considered sequence fragment). This means that the respective sampling distributions are defined by the inside and outside values derived in step one (providing information on the distribution of all possible choices according to the actual input sequence) and the grammar parameters (transition probabilities).

Since every of the before mentioned conditional distributions needed for randomly drawing one of the respective possible choices can be derived in linear time (during the sampling process), any *valid*[4] base pair can be sampled in time $O(n)$. Thus, since any structure of size n can have at most $\lfloor \frac{n-\min_{HL}}{2} \rfloor$ base pairs, a random candidate structure for the given input sequence can be generated in $O(n^2)$ time.

Thus, one straightforward approach for improving the performance of the overall sampling algorithm in the worst-case is to reduce the $O(n^3)$ time complexity required for the preprocessing step at least to the quadratic time of the sampling strategy. To us, this means we might be able to save a significant amount of time by replacing the exact inside-outside calculations with a corresponding heuristic method yielding only approximative inside-outside values for a given input sequence. To see if this might actually be successful, we next want to determine to which extend the inside and outside probabilities react to different types and degrees of disturbances in order the get evidence if it could actually be possible to find an appropriate heuristic.

2.2 Disturbance Types and Levels

We decided to disturb the exact inside and outside probabilities for a given input sequence r of length n in the following ways: For each $X \in I_{\mathcal{G}_s}$ and $1 \leq i, j \leq n$, redefine the corresponding inside value according to

$$\alpha_X(i,j) := \max(\min(\alpha_X(i,j) + \alpha_{err}, 1), 0), \qquad (3)$$

where α_{err} is randomly chosen from the following interval or set:

$$[-\max_{\mathrm{ErrPerc}}\alpha_A(i,j), +\max_{\mathrm{ErrPerc}}\alpha_A(i,j)] \text{ or}$$
$$\{-\mathrm{fix}_{\mathrm{ErrPerc}}\alpha_A(i,j), +\mathrm{fix}_{\mathrm{ErrPerc}}\alpha_A(i,j)\}$$

(relative errors), with $\max_{\mathrm{ErrPerc}}, \mathrm{fix}_{\mathrm{ErrPerc}} \in (0,1]$ defining percentages, or else,

$$[-\max_{\mathrm{ErrVal}}, +\max_{\mathrm{ErrVal}}] \text{ or } \{-\mathrm{fix}_{\mathrm{ErrVal}}, +\mathrm{fix}_{\mathrm{ErrVal}}\}$$

(absolute errors), with $\max_{\mathrm{ErrVal}}, \mathrm{fix}_{\mathrm{ErrVal}} \in (0,1]$ being fixed values. Random errors on all outside values $\beta_X(i,j)$, $X \in I_{\mathcal{G}_s}$ and $1 \leq i, j \leq n$, can be generated in the same way.

The needed conditional sampling distributions (as considered by a particular strategy) are then derived from the exact grammar parameters and the disturbed inside-outside probabilities for the input sequence. This might create the need to (slightly) modify a particularly employed sampling strategy for being capable of dealing with these skewed distributions, as we will see in Section 4.1.

2.3 Analysis of Disturbance Influence

To get a first impression on the influence of disturbances (in the ensemble distribution for a given input sequence) on the quality of generated sample sets, we opted for the potentially most intuitive application in this context, namely *probability profiling* for unpaired bases within particular loop types (see, e.g., (Ding and Lawrence, 2003)). In principle, for each nucleotide position i, $1 \leq i \leq n$, of a given sequence of length

[4]One may for example consider only the 6 different most stable *canonical* pairs as valid ones (like usually done in physics-based approaches due to missing thermodynamics parameters for *non-canonical* pairs). However, we decided to drop this restriction, considering all possible noncrossing base pairings to be valid.

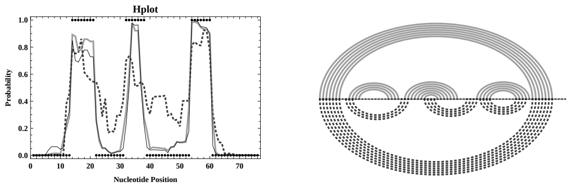

Figure 1: Hairpin loop profile and MP prediction obtained for *E.coli* tRNA^{Ala}. All results have been derived from samples of size 1,000, generated with $\min_{\text{hel}} = 2$ and $\min_{HL} = 3$. Errors were produced with $\max_{\text{ErrPerc}} = 0.99$ (thick gray lines) and $\text{fix}_{\text{ErrPerc}} = 0.99$ (thick dotted darker gray lines).The profiles also display the respective exact results (thin black lines) and the native folding of *E.coli* tRNA^{Ala} (black points).

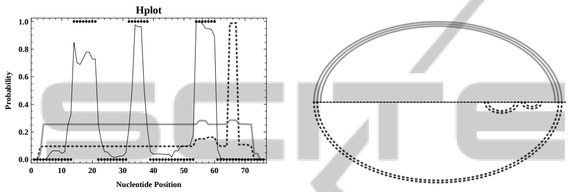

Figure 2: Sampling results for *E.coli* tRNA^{Ala} corresponding to those presented in Figure 1, where $\max_{\text{ErrVal}} = 10^{-9}$ (thick gray lines) and $\text{fix}_{\text{ErrVal}} = 10^{-9}$ (thick dotted darker gray lines) have been chosen for generating the disturbances.

n, one computes the probabilities that i is an unpaired base within a specific loop type. These probabilities are given by the observed frequencies in a representative statistical sample of the complete ensemble (of all possible secondary structures) for the given input sequence.

Furthermore, in order to investigate to what extend the accuracy of predicted foldings changes when different dimensions of relative disturbances are incorporated into the needed sampling probabilities, we will additionally derive the most probable (MP) structure in the generated samples, respectively, as prediction. Note that for our examinations, we will exemplarily consider a well-known trusted tRNA structure, *Escherichia coli* tRNA^{Ala}, since this molecule folds into the typical cloverleaf structure, making it very easy to judge the accuracy of the resulting profiles and predictions.

Figure 1 indicates that even in the case of large relative errors, the sampled structures still exhibit the typical cloverleaf structure of tRNAs, especially for the extenuated disturbance variant according to $\max_{\text{ErrPerc}}$ which seems to have practically no effect on the resulting sampling quality and prediction accuracy. However, Figure 2 perfectly demonstrates that if the disturbances have been created by generating absolute errors on all inside values, then – even for rather small values – the resulting samples (and corresponding predictions as well) seem to be useless. Nevertheless, it seems reasonable to believe that the inside and outside probabilities do not necessarily have to be computed in the exact way, but it may probably suffice to only (adequately) approximate them.

3 HEURISTIC PREPROCESSING

According to the previous discussion, the proclaimed aim of this section is to lower the $O(n^3)$ time complexity for preliminary inside-outside calculations to $O(n^2)$, such that the preprocessing has the same worst-case time requirements as the subsequent sampling process (for constructing a constant number of random secondary structure of size n).

3.1 Basic Idea

The main idea for reaching this time complexity reduction by a factor n in the worst-case is actually quite simple: Instead of deriving the inside values $\alpha_X(i,j)$ (and the corresponding outside probabilities $\beta_X(i,j)$), $X \in I_{\mathcal{G}_s}$, for any combination of start position i and end position j, $1 \leq i,j \leq n$, we abstract from the actual position of subword $R_{i,j} = r_i \dots r_j$ in the input sequence and consider only its length $d = |r_i \dots r_j|$. Thus, for any $X \in I_{\mathcal{G}_s}$, we do not need to calculate $O(n^2)$ values $\alpha_X(i,j)$ (and $\beta_X(i,j)$) for $1 \leq i,j \leq n$, but only $O(n)$ values $\alpha_X(d)$ (and $\beta_X(d)$) for $0 \leq d \leq n$. However, the problem with this approach is that distance d alone may be associated with any of the strings in $\{r_i \dots r_j \mid j-i+1 = d\}$, i.e. without using positions i and j we are inevitably forced to additionally abstract from the actual input sequence r.

Note that it is also possible to combine both alternatives, that is we can first use the traditional algorithms to calculate exact values $\alpha_X(i,j)$ (and $\beta_X(i,j)$) within a window of fixed size W_{exact}, i.e. for $j-i+1 \leq \mathrm{W}_{exact}$ (and $j-i+1 \geq n-\mathrm{W}_{exact}$), and afterwards derive the remaining values for $\mathrm{W}_{exact} < d \leq n$ (and $0 \leq d < n-\mathrm{W}_{exact}$) in an approximate fashion by employing the time-reduced variant for obtaining $\alpha_X(d)$ (and $\beta_X(d)$) for each $X \in I_{\mathcal{G}_s}$. Since W_{exact} is constant, this effectively yields an improvement in the time complexity of the corresponding complete inside computation, which is then given by $O(n^2 \cdot \mathrm{W}_{exact})$. However, even for fix W_{exact} the time requirements for such a mixed outside computation are $O(n^3)$.

3.2 Approximative Emission Terms

Due to the unavoidable abstraction from sequence, we have to determine some approximated terms for the emissions of unpaired bases and base pairs, respectively, that

- do not depend on the positions of subwords within the overall input word, but
- should at least depend on the lengths of the corresponding subwords,

where it is strongly recommended to make sure that as much information on the composition of the actual input sequence as possible is incorporated into these approximated terms.

Therefore, we decided to use the following emission terms that incorporate relative frequencies $\mathrm{rf}^1_{em}(r_i, i-i+1)$ and $\mathrm{rf}^2_{em}(r_i r_j, j-i+1)$ for unpaired bases and base pairs, respectively, that can be efficiently derived from the actual input sequence:

$$\widehat{\Pr}^1_{em}(1) := \sum_{u \in \Sigma_{\mathcal{G}_r}} \Pr^1_{em}(u) \cdot \mathrm{rf}^1_{em}(u,1), \tag{4}$$

$$\widehat{\Pr}^2_{em}(d) := \sum_{p_1 p_2 \in \Sigma^2_{\mathcal{G}_r}} \Pr^2_{em}(p_1 p_2) \cdot \mathrm{rf}^2_{em}(p_1 p_2, d). \tag{5}$$

3.3 (Improved) Approximated Sampling Probabilities

Fortunately, during the complete sampling process, not only the start and end positions of the currently considered sequence fragment $R_{i,j}$, $1 \leq i, j \leq n$, but also the actual input sequence r are always known. Thus, we can in certain cases easily remove some approximate factors in the corresponding approximated inside and outside probabilities and replace them with the respective correct terms (depending on i, j and r) in order to obtain more reliable values.

Therefore, for any sampling strategy, the sampling probabilities from which the respective (conditional) distributions for possible choices are inferred should be defined by using such improved inside and outside probabilities (instead of the corresponding uncorrected precomputed ones). For example, if $X \in I_{\mathcal{G}_s}$ generates hairpin loops, we should use

$$\widehat{\alpha}_X(i,j) := \begin{cases} \alpha_X(i,j), \text{ if } (j-i+1) \leq \mathrm{W}_{exact}, \\ \alpha_X(j-i+1) \cdot \mathrm{c}^1_{em}(i,j), \text{ else}, \end{cases} \tag{6}$$

and

$$\widehat{\beta}_X(i,j) := \begin{cases} \beta_X(i,j), \text{ if } (j-i+1) \geq n - \mathrm{W}_{exact}, \\ \beta_X(j-i+1) \times \\ \mathrm{c}^2_{em}(i-\min_{\mathrm{hel}}, j+\min_{\mathrm{hel}}, \min_{\mathrm{hel}}), \text{ else}, \end{cases} \tag{7}$$

where

$$\mathrm{c}^1_{em}(s,e) := \frac{\prod_{k=s}^{e} \Pr^1_{em}(r_k)}{\widehat{\Pr}^1_{em}(1)^{e-s+1}} \tag{8}$$

and

$$\mathrm{c}^2_{em}(i,j,l) := \frac{\prod_{k=0}^{l-1} \Pr^2_{em}(r_{i+k} r_{j-k})}{\prod_{k=0}^{l-1} \widehat{\Pr}^2_{em}((j-k)-(i+k)+1)}. \tag{9}$$

4 CONSIDERED SAMPLING STRATEGIES

For the subsequent examinations, we will employ two different sampling strategies, which are introduced now.

4.1 Well–Established Strategy

Let us first consider a slightly modified variant of the rather simple and widely known sampling strategy from (Ding and Lawrence, 2003; Nebel and Scheid, 2011). Briefly, this well-established strategy samples a complete secondary structure $S_{1,n}$ for a given input sequence r of length n in the following recursive way: Start with the entire RNA sequence $R_{1,n}$ and consecutively compute the adjacent substructures (single-stranded regions and paired substructures) of the exterior loop (from left to right). Any (paired) substructure on fragment $R_{i,j}$, $1 \leq i < j \leq n$, is folded by recursively constructing substructures (hairpins, stacked pairs, bulges, interior and multibranched loops) on smaller fragments $R_{l,h}$, $i \leq l < h \leq j$. That is, fragments are sampled in an *outside-to-inside* fashion.

Notably, without disturbances of the underlying probabilistic model, it is guaranteed that any sampled loop type for a considered sequence fragment can be successfully generated (otherwise its probability would have been 0). As this must not hold in disturbed cases (like e.g. those of Section 2.3), the most straightforward modification to solve this problem is that in any such case where the chosen substructure type can not be successfully generated, the strategy returns the partially formed substructure. Figure 3 gives a schematic overview on this inherently controlled sampling strategy.

As regards this particular sampling strategy, the outside values can easily be omitted from the corresponding formulae for defining the needed sampling probabilities, since in any case they contribute the same multiplicative factor to the distinct sampling probabilities for mutually exclusive and exhaustive cases, such that they finally do not influence the sampling decision at all.

The correctness of this simplification can easily be verified by considering a particular set $ac_X(i,j)$ of all choices for (valid) derivations of intermediate symbol

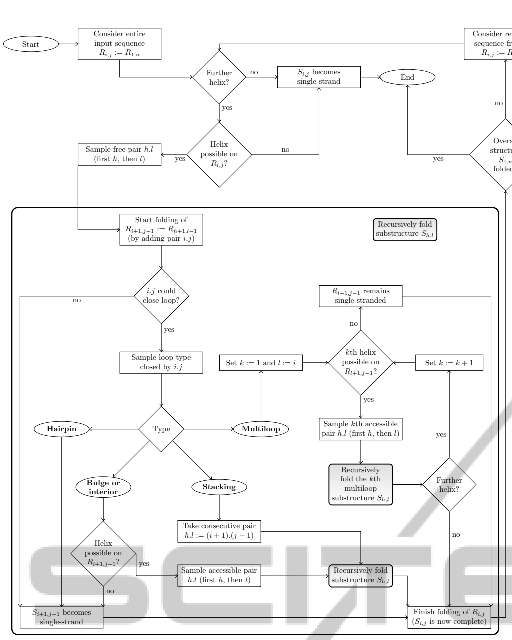

Figure 3: Flowchart for recursive sampling of an RNA secondary structure $S_{1,n}$ for a given input sequence r of length n according to an inherently controlled strategy with predetermined order, similar to that of (Ding and Lawrence, 2003; Nebel and Scheid, 2011)).

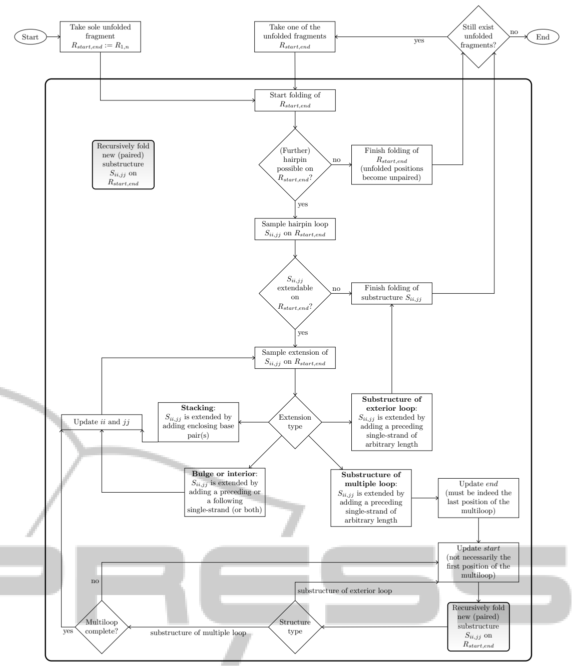

Figure 4: Flowchart for recursive sampling of an RNA secondary structure $S_{1,n}$ for a given input sequence r of length n according to a less restrictive strategy with extensively more freedom (that requires dynamic validation of possible random choices during the sampling process).

$X \in I_{\mathcal{G}_s}$ on sequence fragment $R_{i,j}$, $1 \leq i < j \leq n$, which actually correspond to possible substructures on $R_{i,j}$. Under the assumption that the alternatives for intermediate symbol X are $X \to Y$ and $X \to VW$, the (valid) mutually exclusive and exhaustive cases are defined by:

$$acX(i,j) := acX_Y(i,j) \cup acX_{VW}(i,j), \tag{10}$$

where

$$acX_Y(i,j) := \{\{0,p\} \mid p \neq 0 \text{ for } p = \beta_X(i,j) \cdot \alpha_Y(i,j) \cdot \Pr^0_{tr}(X \to Y)\} \tag{11}$$

and

$$acX_{VW}(i,j) := \{\{k,p\} \mid i \leq k \leq j \text{ and } p \neq 0 \text{ for } p = \beta_X(i,j) \cdot \alpha_V(i,k)\alpha_W(k+1,j) \cdot \Pr^0_{tr}(X \to VW)\}. \tag{12}$$

We then sample from the probability distribution induced by $acX(i,j)$ (conditioned on fragment $R_{i,j}$), which implies

$$1 = \sum_{p \in acX(i,j)} \frac{p}{z} = \frac{\beta_X(i,j)}{z} \sum_{p \in acX(i,j)} \frac{p}{\beta_X(i,j)} \tag{13}$$

for $z := \sum_{p \in acX(i,j)} p$, since $\beta_X(i,j) \neq 0$ due to the definition of $acX(i,j)$.

4.2 Alternative Strategy

Unfortunately, the common sampling strategy from Section 4.1 lacks the ability to take full advantage of the exact inside values $\widehat{\alpha}_X(i,j) = \alpha_X(i,j)$, for $X \in I_{\mathcal{G}_s}$ and $j-i+1 \leq \mathrm{W}_{exact}$, obtained by employing a particular mixed preprocessing variant according to $0 \leq \mathrm{W}_{exact} < n$. Particularly, the strategy in general inevitably has to sample the first base pairs from corresponding conditional probability distributions for rather large fragments $R_{i,j}$ with $j-i+1 > \mathrm{W}_{exact}$, which are indeed induced by approximated sampling probabilities rather than exact ones. Therefore, we worked out an alternative to this well-established sampling strategy that obeys to contrary principles, resulting in a reverse sampling direction.

Basically, a complete secondary structure $S_{1,n}$ for a given input sequence r of length n can alternatively albeit unconventionally be sampled in the following (deliberately less controlled) way: Start with the entire RNA sequence $R_{start,end} = R_{1,n}$ and randomly construct adjacent substructures (paired substructures preceeded by potentially empty single-stranded regions) of the exterior loop on the considered sequence fragment $R_{start,end}$ (where the construction does not follow a particular order, e.g. does not sample from left to right), as long as no further paired substructure can be folded. Any (paired) substructure on fragment $R_{start,end}$, $1 \leq start \leq end \leq n$, is created by sampling a random hairpin loop (with closing base pair $i.j$, for $start < i < j < end$) – here we can take advantage of

Table 1: Comparsion of the considered sampling strategies (for an arbitrary input sequence of length n).

Aspect	Conventional Strategy	Alternative Strategy
Preprocessing time	$O(n^3)$ for exact calculations, $O(n^2)$ for approximate variant, $O(n^2)$ with constant $W_{exact} \geq 0$	$O(n^3)$ for exact calculations, $O(n^2)$ for approximate variant, $O(n^3)$ with constant $W_{exact} \geq 0$
Constraints	None	Constant $\max_{hairpin}$, $\max_{bulge}$ and $\max_{strand}$
Characteristics and course of action	Inherently controlled, ordered: - substructures from left to right, - sampling proceeds "inwards": construction of substructure $S_{i,j}$ starts by considering $R_{i,j}$ and ends by generating an unpaired region (usually a hairpin loop)	Extensively more freedom, less restrictive: - substructures in arbitrary order, - sampling proceeds "outwards": construction of new substructure on unfolded fragment $R_{start,end}$ starts with random hairpin loop which is extended to a complete and valid (paired) substructure $S_{i,j}$ on $R_{start,end}$
Benefits of sampling direction	(Sub)structures are folded in accordance with the generation of the corresponding (unique leftmost) derivation (sub)tree by the underlying SCFG	Takes more advantage of inside probabilities for shorter fragments containing less approximated terms and thus less inaccuracies (although this potential is narrowed by the outside values for which the contrary holds)
Function of outside values	Not considered (do not influence sampling distributions)	1) "Normalize" sampling probabilities 2) Ensure valid extensions
Identification of valid choices	Not required (all possible choices are principally valid)	Dynamic checking required (due to dependence on previously folded substructures)
Folding time	$O(n^2)$	$O(n^2)$ with larger constants
Overall time complexity for MP predictions	$O(n^3)$ with exact variant, $O(n^2)$ with constant $W_{exact} \geq 0$ or in completely approximated case	$O(n^3)$ in case of exact computations or mixed variants according to $W_{exact} \geq 0$, $O(n^2)$ only in completely approximated case

exact inside values from a mixed preprocessing since most likely i and j are close – and extending it (towards the ends of $R_{start,end}$) by successively drawing closing base pairs. During this extension, basically all known substructures (stacked pairs, bulges, interior and multibranched loops, that obey to certain restrictions which will be discussed later) may be folded, where each substructure (e.g. multiloop) has to be completed before its closing base pair is added and the corresponding helix can actually be further extended. The process of folding a particular paired substructure ends with a complete and valid paired structure (of the currently folding multiloop or of the exterior loop), either with or without a directly preceeding unpaired region, both on the considered fragment $R_{start,end}$. Figure 4 gives a schematic overview on this *inside-out* fashion sampling strategy.

Note that in order to ensure that all sampled substructures can be successfully folded, especially in the case of multiloops, we have to take care that at any point, the strategy may only draw such random choices that do not make it impossible to successfully finish the currently running construction process (of a particular loop). As this strongly depends on the actual positions and types of all previously folded paired substructures, the algorithm obviously needs to dynamically determine the respective set of all *valid* choices (during the sampling process itself) before a corresponding probability distribution (needed for drawing a particular random choice) can be derived.

This, however, may cause severe problems as regards the time complexity for randomly sampling the next extension (or base pair). Nevertheless, in order to guarantee that the worst-case time complexity for drawing any random choice remains in $O(n)$, we only need to impose a few restrictions concerning the lengths of single-stranded regions in some types of loops. In detail, we have to consider a maximum allowed number of nucleotides in unpaired regions of hairpin loops ($\max_{hairpin}$), bulge or interior loops ($\max_{bulge}$), and multiloops ($\max_{strand}$)[5]

For example, if $X \in I_{G_s}$ generates hairpin loops, then the set of all possible hairpin loops that can be validly folded on sequence fragment $R_{start,end}$ is given by

$$\begin{aligned}
&\mathrm{pcHL}(start, end) := \big\{\{i, j, p\} \mid \\
&\quad start + \min_{\mathrm{hel}} \leq i \leq j \leq end - \min_{\mathrm{hel}} \text{ and} \\
&\quad i + \min_{HL} - 1 \leq j \leq i + \max_{hairpin} - 1 \text{ and} \\
&\quad R_{i-\min_{\mathrm{hel}}, j+\min_{\mathrm{hel}}} \text{ not folded and} \\
&\quad p = \widehat{\beta}_X(i,j) \cdot \widehat{\alpha}_X(i,j) \neq 0\big\}. \qquad (14)
\end{aligned}$$

Obviously, $\max_{hairpin}$ indeed ensures that

[5]Note that these restrictions are not as severe as it may seem, since for example choosing the constant value 30 (as also done by many MFE based prediction algorithms) for all three parameters can be expected to hardly have a negative impact on the resulting sampling quality.

pcHL$(start, end)$ can be computed in $O(n)$ time.

Finally, it should be noted that this sampling strategy needs to additionally consider outside probabilities, for two reasons: First, for "normalizing" the resulting sampling probabilities. This is due to the fact that the different possible choices $\{i, j, p\}$ usually imply substructures $S_{i,j}$ of different lengths $j-i+1$, such that only $p = \widehat{\alpha}_X(i,j) \cdot \widehat{\beta}_X(i,j)$ ensures that the probabilities of all possible choices are of the same order of magnitude and hence imply a reasonable probability distribution for drawing a random choice. Second, the outside values are required for guaranteeing that sampled substructures can be validly extended. This means that only such hairpin loops and extensions (implying a surrounding base pair $i.j$) may be sampled that can actually lead to the generation of a corresponding valid helix.

We conclude this section by referring to Table 1 that summarizes the main differences of both sampling variants.

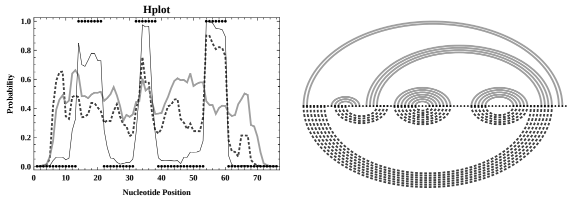

Figure 5: Sampling results for *E.coli* tRNA^{Ala}, derived with the common strategy (under the assumption of $\min_{\text{hel}} = 2$ and $\min_{HL} = 3$), where we used sample size 100,000, 10,000 and 1,000 for $W_{exact} = -1$ (no window, thick gray lines), $W_{exact} = 30$ (moderate window, thick dotted darker gray lines) and $W_{exact} = +\infty$ (complete window, thin black lines), respectively.

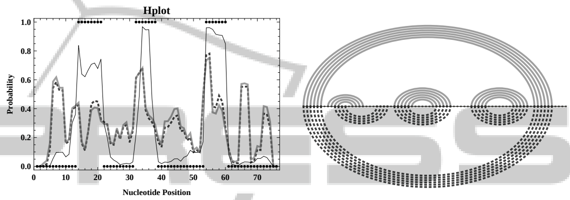

Figure 6: Sampling results corresponding to those of Figure 5, obtained by employing the alternative sampling strategy.

5 APPLICATIONS

First, the sampling results shown in Figure 5 indicate that for the common sampling strategy, considering a window of constant size W_{exact} (chosen to cover the size of hairpin-loops) with a mixed preprocessing variant, actually yields a slight improvement of the resulting sampling quality, where the same time requirements are needed for generating the respective sample sets.

Contrary to this observation, Figure 6 demonstrates that when employing our alternative sampling strategy, the corresponding results are not significantly different for the completely approximate preprocessing variant and for a mixed version on the basis of a constant value for W_{exact}. Thus, to our surprise it does not matter if we consider a constant window for exact calculations or simply approximate all inside and outside values, which is not only an interesting observation itself, but also fortunately prevents us from having to deal with an undesirable trade-off between reducing the worst-case time complexity (by a linear factor) and sacrificing less of the resulting sampling quality. In fact, this means we may (without resulting significant quality losses) always use the more efficient approximative preprocessing variant in order to reduce the worst-case time complexity of the overall sampling algorithm.

However, all profiles perfectly demonstrate that due to the noisy ensemble distribution caused by approximating the highly relevant sequence-dependent emission probabilities, the resulting sample sets usually contain many foldings that are rather unlikely according to the exact distribution for the considered input sequence. For this reason, it can not be recommended to employ one of the following otherwise reasonable construction schemes for deriving predictions according to the entire sample set: we should rather neither predict γ-MEA nor γ-centroid structures of the generated sample set as defined in (Nebel and Scheid, 2011), since those effectively reflect the overall behavior of the sample set. Those predictions must anyway be considered inappropriate choices in our case, since their computation requires $O(n^3)$ time, which would inevitably undo the time reduction reached by approximating. Nevertheless, we can without significant losses in performance (without increasing the worst-case time complexity of the overall algorithm) identify the MP structure of the generated sample[6], in strong analogy to traditional SCFG approaches. Since for this selection principle, we can actually rely on the exact distribution of feasible structures[7], this seems to

[6]The probability of each structure can either be determined on the fly while sampling, multiplying the probabilities of the production rules which correspond to the respective sampling decisions, and otherwise – since the underlying SCFG from (Nebel and Scheid, 2011) is unambiguous – are computable in $O(n^2)$ time making use, e.g., of an Earley-style parser.

[7]Note that the probability for a particular folding of a given RNA sequence is equal to a product of (different pow-

be the right choice indeed.

On the basis of a series of experiments, we observed that stability in resulting predictions and a competitive prediction accuracy can only be reached by increasing the sample size, especially in the case of complete approximation for the preprocessing step and sampling according to the alternative strategy introduced in Section 4.2. That is, more candidate structures ought to be generated for guaranteeing that the resulting MP predictions are reproducible (by independent runs for the same input sequence) and of hight quality. This negative effect is considerably lowered by using (larger) constant values of $W_{exact} \geq 0$, and is actually less recognizable when employing the conventional sampling strategy recapped in Section 4.1.

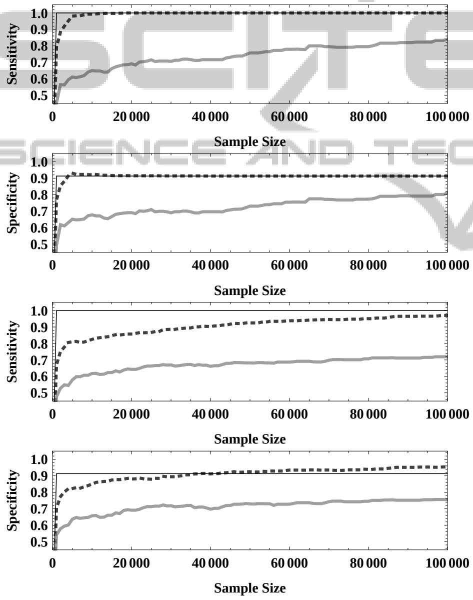

Figure 7: Sensitivity and specificity of predictions as a function of sample size derived for *E.coli* tRNA^{Ala}. Top (bottom) lines show the common (alternative) sampling strategy.

Figure 7 shows the averaged sensitivities and specificities obtained for 50 independent runs of continuously sampling secondary structures taking the so far most probable one as the actual prediction (which determines sensitivity and specificity for the actual sample size). We observe that when making use of approximate probabilities sample sizes about 40 to 50 times as large as for a precise preprocessing are needed to generate competitive predictions. Thus, for a naive implementation the speedup gained by approximation may partly be lost. However, unlike prediction algorithms using dynamic programming, sampling can easily be parallelized. Making use of a grid environment where today one may assume a processor to have about 8 cores, a grid of size 5 or 6 computers is sufficient to compensate the increased sample size. Furthermore, since we only make use of the most probable sampled structure for our prediction, sampling can be done in-place, storing in each core only the best structure seen so far. This reduces the memory requirements and keeps the communication costs rather moderate since it is finally only necessary to gather m structures from m cores and select the best. We performed a series of experiments, making use of Mathematica's parallel computation features, which proved that the overall process scales linearly in the number of cores used with a non-measurable communication overhead. This finally proves the applicability of our approach providing a factor n speedup compared to established prediction tools but still maintaining the limits implied by a quadratic memory consumption (in our case used to store parameter values).

6 CONCLUSIONS

The major advantage of the presented approximative method is that it is more efficient than all other modern prediction algorithms (implemented in popular tools like Mfold (Zuker, 2003), Vienna RNA (Hofacker, 2003), Pfold (Knudsen and Hein, 2003), Sfold (Ding et al., 2004) or CONTRAfold (Do et al., 2006)), reducing the worst-case time complexity by a linear factor, such that the time and space requirements are both bounded by $O(n^2)$. However, a potential drawback lies in the observation that the overall quality of generated samples decreases (as indicated by probability profiling for specific loop types), which is due to the approximated ensemble distribution. As a consequence, we usually need to use larger sample sizes for obtaining a competitive prediction accuracy and stable predictions, i.e., more candidate structures for a given input sequence have to be generated to ensure that the approximation method outputs rather identical predictions in independent runs for that sequence. According to our experiments, an efficient implementation that really takes advantage of the accelerated preprocessing (3.7 compared to 49 seconds for our proof-of-concept implementation in Wolfram Mathematica) but handles large sample sizes can be obtained by parallelization.

Note that all results presented in this article have been

ers of the diverse) transition and emission probabilities (according to the corresponding derivation tree), which means it depends only on the exact trained parameter values of the underlying SCFG.

derived with a purposive proof-of-concept implementation of the described methods. A more sophisticated tool will be realized in the future, hoping that the proposed prediction approach proves capable of yielding acceptable accuracies even for such types of RNAs whose molecules imply a great variety of structural features (due to large sequence lengths). In fact, we here only considered exemplary applications for one particular tRNA molecule in order to get positive feedback that (at least) the MP predictions obtained via approximated SCFG based sampling can be of high quality. Accordingly, more general experiments are needed, e.g., in connection with RNA molecules of sizes $n = 3000 - 30000$ (for which the memory constraints of our approach are not restrictive assuming 1GB of memory for each core) and where long distance base pairs in a global folding are of interest. In such a scenario the proposed algorithm could be the method of choice provided it performs similarly well.

This line of research is work in progress, but we found the first impressions presented within this note so motivating that we wanted to share them with the scientific community already at this point, primarily because this work leaves a number of open questions that may be inspiration for further research of other groups. For instance, recall that we used a sophisticated SCFG (representing a formal language counterpart to the thermodynamic model applied in the Sfold program) as probabilistic basis for the considered sampling strategies. However, it would also be possible to employ other SCFG designs, for example one of the commonly known *lightweight* grammars from (Dowell and Eddy, 2004). This might of course yield at least noticeable if not significant changes in the resulting sampling quality, which could be an interesting subject to be explored.

It should also be noted that a similar approximative approach could potentially be considered when attempting to reduce the worst-case time complexity of the sampling extension of the PF approach. In fact, since sequence information is incorporated into the used (equilibrium) PFs and corresponding sampling probabilities only in the form of particular sequence-dependent free energy contributions, it seems reasonable to believe that the time complexity for the forward step (preprocessing) could possibly be reduced by a linear factor to $O(n^2)$ when using some sort of approximated (averaged) free energy contributions that do not depend on the actual sequence (but contain as much sequence information as possible), in analogy to the approximated preprocessing step (inside and outside calculations) considered in this work, where we eventually only had to use averaged emission terms instead of the exact emission probabilities in order to save time.

REFERENCES

Akutsu, T. (1999). Approximation and exact algorithms for RNA secondary structure prediction and recognition of stochastic context-free languages. *J. Comb. Optim.*, 3(2–3):321–336.

Backofen, R., Tsur, D., Zakov, S., and Ziv-Ukelson, M. (2011). Sparse RNA folding: Time and space efficient algorithms. *Journal of Discrete Algorithms*, 9:12–31.

Ding, Y., Chan, C. Y., and Lawrence, C. E. (2004). Sfold web server for statistical folding and rational design of nucleic acids. *Nucleic Acids Research*, 32:W135–W141.

Ding, Y. and Lawrence, C. E. (2003). A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Research*, 31(24):7280–7301.

Do, C. B., Woods, D. A., and Batzoglou, S. (2006). CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, 22(14):e90–e98.

Dowell, R. D. and Eddy, S. R. (2004). Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5:71.

Frid, Y. and Gusfield, D. (2010). A simple, practical and complete $O(n^3/\log(n))$-time algorithm for RNA folding using the Four-Russians speedup. *Algorithms for Molecular Biology*, 5(1):5–13.

Hofacker, I., Fontana, W., Stadler, P., Bonhoeffer, S., Tacker, M., and Schuster, P. (1994). Fast folding and comparison of rna secondary structures (the Vienna RNA package). *Monatsh Chem.*, 125(2):167–188.

Hofacker, I. L. (2003). The vienna RNA secondary structure server. *Nucleic Acids Research*, 31(13):3429–3431.

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

Knudsen, B. and Hein, J. (2003). Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research*, 31(13):3423–3428.

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

Nebel, M. E. and Scheid, A. (2011). Evaluation of a sophisticated SCFG design for RNA secondary structure prediction. Submitted.

Wexler, Y., Zilberstein, C., and Ziv-Ukelson, M. (2007). A study of accessible motifs and RNA folding complexity. *Journal of Computational Biology*, 14(6):856–872.

Zuker, M. (1989). On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52.

Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.*, 31(13):3406–3415.