On a Statistical Filter for RNA Secondary Structures

Markus E. Nebel*
Johann Wolfgang Goethe-Universität
Institut für Informatik
Robert Mayer Str. 11-15
60325 Frankfurt am Main
Germany

Abstract

Predicting the secondary structure of RNA molecules from the knowledge of the primary structure (the sequence of bases) is still a challenging task. However, many efforts have been made over the last decades. For instance, the algorithm of Zuker [30] is able to give good results based on the search of an energetic optimal configuration. Nevertheless the output of such algorithms does not always provide the real folding of the molecule and it would be a nice feature to cross-check them with well-known structures of the same type of RNA. In this paper we show how to use probability generating functions as a model for RNA secondary structures. These generating functions are derived from stochastic context-free grammars that are trained on a database of RNA secondary structures, i.e. their probabilities are adapted to the real world data. The resulting model is very realistic and can be used to derive results concerning the average shape of a molecule together with the corresponding variances and higher moments. We propose to use this information as a statistical filter for structures predicted by any algorithm: If a number of conserved structural parameters, i.e. parameters with a small variation, are too far beyond the expectations one should not rely on the prediction.

1 Introduction and Basic Definitions

A ribonucleic acid (RNA) molecule consists of a chain of nucleotides of which exist four different types. Each nucleotide contains a base, a phosphate group and a sugar group. The various types of nucleotides only differ by the base involved; there are four choices for the base namely adenine (A), cytosine (C), guanine (G) and uracil (U). The specific sequence of the bases along the chain is called *primary structure* of the molecule. It is usually modeled as a word over the alphabet $\{A, C, G, U\}$. Through the creation of hydrogen bounds, the complementary bases A and U (resp. C and G) form stable base pairs with each other. Additionally, there is the weaker G-U pair, where bases bind in a skewed fashion. Due to these base pairs, the linear chain is folded into a three-dimensional conformation called *tertiary structure* of the molecule. For some types of RNA molecules like for example transfer RNA, the tertiary structure is highly connected with the function of the molecule. Since experimental approaches which allow the discovery of the tertiary structure are quite expensive biologists are looking for methods which make it possible to predict the tertiary structure from the knowledge of the primary structure. With respect to this concern it is customary to consider the simplified *secondary structure* of the molecule, where we restrict the possible base pairs such that only planar structures occur. Over the last decades many efforts have been made about the prediction of the secondary structure and several algorithms using rather different ideas were presented [5, 18, 21, 22, 25, 28, 30]. However, the output of such algorithms should not be

*nebel@sads.informatik.uni-frankfurt.de

assumed to be error-free so sometimes they predict a wrong folding of a molecule. Thus it would be a nice feature to cross-check them with well-known structures of the same type of RNA. In this paper we propose to use a statistical filter for this purpose which compares structural parameters of the predicted molecule with those of an *expected molecule* of the same type and the same size (number of nucleotides/bases), and we show how such a filter can be computed. If a number of conserved structural parameters, i.e. parameters with a small variation, are too far beyond the expectations one should not rely on the prediction. Literature offers a lot of different results dealing with the expected structure of RNA molecules. Starting with the pioneering work of Waterman [28] in which the first formal framework for secondary structures was given, authors considered the combinatorial and the Bernoulli model of RNA secondary structures (where the molecule is modeled as a certain kind of planar graph) and they derived numerous results like the average size and number of hairpins and bulges, the number of ladders, the expected order of a structure and its distribution or the distribution of unpaired bases (see [14, 11, 19, 20]). In [20] it was pointed out that both models are rather unrealistic and thus the corresponding results can hardly be used for our purposes. We show how it is possible to construct a realistic model for RNA secondary structures which allows us to derive the corresponding expectations, variances and all other higher moments to be used according to our ideas. In the rest of this paper we assume that the reader is familiar with the basic notions of Formal Language Theory such as context-free grammars, derivation trees, etc. A helpful introduction to the theory can be found in [15]. We also assume a working knowledge on the notion of secondary structures and the concepts like hairpins, interior loops, etc. We refer to [26, Ch. 3] for a related introduction.

Besides modelling a secondary structure as a planar graph, it is a slightly different approach to model it by using stochastic context-free grammars as proposed by [24]. A stochastic context-free grammar (SCFG) is a 5-tuple $G = (I, T, R, S, P)$, where I (resp. T) is an alphabet (finite set) of intermediate (resp. terminal) symbols (I and T are disjoint), $S \in I$ is a distinguished intermediate symbol called *axiom*, $R \subset I \times (I \cup T)^\star$ is a finite set of production-rules and P is a mapping from R to $[0, 1]$ such that each rule $f \in R$ is equipped with a probability $p_f := P(f)$. The probabilities are chosen in such a way that for all $A \in I$ the equality $\sum_{f \in R} p_f \delta_{Q(f),A} = 1$ holds. Here δ is Kronecker's delta and $Q(f)$ denotes the source of the production f, i.e. the first component A of a production-rule $(A, \alpha) \in R$. In the sequel we will write $p_f : A \to \alpha$ instead of $f = (A, \alpha) \in R$, $p_f = P(f)$. In Information Theory SCFGs were introduced as a device for producing a language together with a corresponding probability distribution (see e.g. [1, 12]). Words are generated as for usual context-free grammars, the product of the probabilities of the used production-rules provides the probability of the generated word. Note that we do not always get a probability distribution for the language in this way. However, there are sufficient conditions which allow to check whether or not a given grammar provides a distribution. One was interested in parameters like the moments of the word and derivation lengths [17] or the moments of certain subwords [6]. Furthermore, one was looking for the existence of standard-forms for SCFGs such as Chomsky normalform or Greibach normalform in order to simplify proofs [16]. Some authors used the ideas of Schützenberger [3] to translate the corresponding grammars into probability generating functions in order to derive their results [6, 17]. However, languages resp. grammars were not used to model any sort of combinatorial object besides languages themselves and therefore the question of how to determine probabilities was not asked. In Computational Biology SCFGs are used as a model for RNA secondary structures [18, 24]. In contrast to Information Theory not only the words generated by the grammar are used, but also the corresponding derivation trees are taken into consideration: A word generated by the grammar is identified with the primary structure of an RNA molecule, its derivation tree is considered as the related secondary structure [24]. Note that there exists a one-to-one correspondence between the planar graphs used by Waterman as a model

for RNA secondary structures and a certain kind of unary/binary trees (see e.g. [19]). Thus the major impact from using SCFGs is given by the way in which probabilities are generated. Since a single primary structure can have numerous secondary structures, an ambiguous SCFG is the right choice. The probabilities of such a grammar can be trained from a database. The algorithms applied for this purpose are generalizations of the forward/backward algorithm used in the context of hidden Markov models [4, 18] and are also applied in Linguistics, where one usualy works with ambiguous grammars, too. At the end of the training the most probable derivation tree of a primary structure in the database equals the secondary structure given by the database. Applications were found in the prediction of RNA secondary structure [5, 18] were the most probable derivation tree is assumed to be the secondary structure belonging to the primary structure processed by the algorithm. So far, no one used these grammars to derive structural results, which in case of an ambiguous grammar is obvious since it is impossible to find any sense in such results. In this paper we provide the link between both disciplines and go even further. We use non-ambiguous stochastic context-free grammars to model the RNA secondary structures. This is done by disregarding the primary structure and representing the secondary structure as a certain kind of Motzkin language, (i.e., a language over the alphabet $\{(,),|\}$, which encodes unary/binary trees equivalent to the secondary structure) which now is the language generated by the grammar. We further propose a simple algorithm to train non-ambiguous SCFGs which works much faster than the algorithms normally used. From the SCFGs we derive probability generating functions which are used to conclude quantitative results related to the structure of RNA secondary structures. In order to train the grammar we derived a database of Motzkin words which correspond one-to-one to the secondary structures contained in the databases of Wuyts et al. [29]. We have also used the databases of Brown for RNase P sequences [2] and of Sprinzl et al. for tRNA molecules [27], the corresponding results are not reported here due to lack of space.

2 A Statistical Filter for Predicted RNA Molecules

In this section we will present our results without any comment on how they were derived; technical details are presented in Section 3. However, we will address possible applications for our findings. Most notably we were able to quantify the expected characteristics of substructures of large subunit (LSU) ribosomal RNA molecules, the corresponding formulæ are presented in Table 1. There each parameter is presented together with its expected asymptotical behavior, i.e., its expected behavior within a large (number of nucleotides) molecule.

Note that we have investigated all the different substructures which must be distinguished in order to determine the total free energy of a molecule which is necessary e.g. for certain predicting algorithms. Compared to all previous attempts to describe the structure of RNA quantitatively (see for instance [14, 19, 20, 23, 28]), the results presented here are the most realistic. They should be considered as the structural behavior of an RNA molecule folded with respect to its energetic optimum. Therefore, they are of interest themselves; for the first time we get some insight on how real secondary structures behave. However, the realistic modelling of the secondary structures gives rise to different applications like for instance the following: Firstly, we can use our results in order to provid bounds for the running-time of algorithms working on secondary structures as their input; we don't want to argue this way. Secondly, when predicting a secondary structure, our results may provide initial values for loop lengths etc. when searching for an optimal configuration such that a faster convergence should be expected. Thirdly, and this is our main concern in this article, the results may serve as a statistical filter for predicted RNA secondary structures. Assume we use some algorithm to predict the secondary structure of a LSU ribosomal RNA sequence. A simple counting program can be used in order to determine parameters like the number of hairpins, bulges, multiloops and so on together with the corresponding sizes (lengths). Setting n within the

Table 1: Expectations for different parameters of large subunit ribosomal RNA secondary structures. In all cases n is used to represent the total size of the molecule.

Parameter	Expectation
Number of hairpins	$0.0226n$
Length of a hairpin-loop	7.3766
Number of bulges	$0.0095n$
Length of a bulge	1.5949
Number of ladders	$0.0593n$
Length of a ladder (counted in the number of basepairs)	4.1887
Number interior loop	$0.0164n$
Length of a single loop within an interior loop	3.8935
Number of multiloop	$0.0106n$
Degree of a multiloop	4.1311
Length of a single loop within a multiloop	4.3686
Number of single stranded regions	18.1679
Length of a single stranded region	18.1353

formulæ of Table 1 to the length of the sequence allows the comparison of the predicted structure to the behavior of an *expected molecule*. So the average structural behavior described by our formulæ serves as some sort of consensus structure. If the predicted structure differs too much for many parameters, we should not rely on the prediction. Besides theory it is necessary to gain experience in order to see whether these ideas work and/or which thresholds are adequate for the different parameters. However, the positive experience of Knudsen et al. [18] and of Eddy et al. [5] with respect to the prediction of secondary structures based on trained SCFGs (resp. covariance models) give rise to be optimistic.

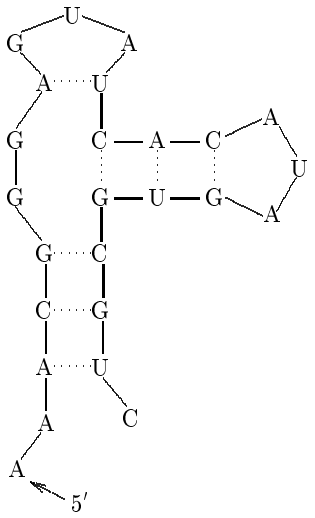
3 Investigating Biological Objects Based on SCFGs

In this section we will describe how the results from the previous section were computed. We will proceed in two steps. First we will discuss a toy example in order to give details of the methodology without stressing the presentation with complex formulæ. Afterwards we will present the key steps into the investigation of LSU ribosomal RNA molecules.

3.1 A Toy Example

Now we will present our method. In some sense it is the combination of several already known concepts from different disciplines which for the first time are used to perform a structural analysis for any kind of combinatorial object (which in our case are the RNA secondary structures). Furthermore it is universal in the sense that it can be used with respect to any type of object which can be represented as a (context-free) formal language. Therefore it should be of independent interest and of relevance not only with respect to the application presented here and thus not only with respect to computational biology. However, since the calculations that are necessary to compute the results for the average shape of LSU ribosomal RNA molecules are rather complicated, we want to explain the method using a small example without real application.

As already mentioned, the secondary structures are modeled as a certain kind of Motzkin words. There, two paired bases are represented as a pair of corresponding brackets $(\cdots)$ somewhere within the word, unpaired bases are represented as $|$. Since by definition, pseudoknots are impossible within a secondary structure, the resulting representation is a correctly bracketed word where symbols $|$ are inserted at appropriate places. Figure 1 shows the graph representation of a secondary



$$w = ||((((|||(|||)(((|||)))))))|$$

Figure 1: The graph representation of a secondary structure and the corresponding Motzkin word.

structure together with its representation as a Motzkin word w. If we forget about restrictions for secondary structures such as for instance a minimal hairpin-loop length of three, the following context-free grammar generates the language of all possible encodings (Motzkin-words):

$$f_1 = S \to S(S), \qquad f_2 = S \to S|, \qquad f_3 = S \to \varepsilon.$$

Here ε is used to denote the empty word, i.e. the application of production $S \to \varepsilon$ deletes symbol S in a sentential form. In order to train the grammars we have extracted from the database of LSU ribosomal RNA secondary structures [29] a file with the corresponding encodings as a Motzkin word. Additionally, we have implemented an Earley-Parser which for all words in the file computes the derivation tree. The probabilities are determined by counting the number of applications of each production-rule f in each derivation tree and dividing it by the total number of applied rules having the source[1] $Q(f)$. The observed probabilities are rounded to the second decimal place in order to keep the toy example as simple as possible. We obtain the probability $\frac{1}{4}$ for f_1 and f_3 and $\frac{1}{2}$ for f_2. By a standard procedure (see e.g. [10]), we translate the grammar together with the probabilities into an equation for the corresponding probability generating function $S(z, y)$. For this purpose we mark each of the symbols in $\{(,), |\}$ by variable z, the symbols $|$ are additionally marked by y which is needed to determine the behavior of unpaired bases. We obtain

$$F(S, z, y) := c_2(z, y)S^2 + c_1(z, y)S + c_0(z, y) = 0$$

with $c_2(z, y) := \frac{1}{4}z^2$, $c_1(z, y) := \frac{1}{2}zy - 1$ and $c_0(z, y) := \frac{1}{4}$ for the defining equation. Note that for polynomial $F(S, z, y)$, S is the variable representing the generating function $S(z, y)$ that we are interested in. In case of such a simple example it would be a child's play to solve the equation for S in order to obtain a closed form representation of the generating function in question. However, for our later applications this would be impossible because of complexity and we thus use a way which will work in those cases, too. We will use the Newton polygon method as described in [10, 13] to find the expansion of our generation function around its dominant singularity from the polynomial $F(S, z, y)$ only. From this expansion it will be possible to derive an asymptotic formula for the coefficient at z^n using the $\mathcal{O}$-transfer method. In order to make this article more self-contained we

[1] Note that the inside/outside algorithm which is normally used to train the grammars has a running time in $\mathcal{O}(n^3)$ while for non-ambiguous grammars our method runs in quadratic time.

first give a brief description of how the $\mathcal{O}$-transfer method works.
Assume we have a generating function $f(z) = \sum_{n\geq 0} f_n z^n$ with $f_n \geq 0$ for all n and we wish to approximate f_n for large n (in our case f_n will be for instance the probability of a secondary structure of size n). In the sequel we will use the notation $[z^n]f(z)$ to denote the coefficient at z^n in the expansion of $f(z)$ around 0. The basic principle of the method is the existence of a correspondence between the asymptotic expansion of $f(z)$ near its dominant singularities and the asymptotic expansion of the coefficients f_n. Here, a singularity is called dominant if it is located on the circle for convergence of $f(z)$ or equivalent if it is a singularity of smallest modulus. It is convenient to consider functions $f(z)$ that are singular at $z = 1$, a restriction that entails no loss of generality: If $f(z)$ is singular at $z = \rho^{-1}$ and $g(z) := f(z/\rho)$, then by the scaling rule of Taylor expansions $[z^n]f(z) = \rho^n[z^n]f(z/\rho) = \rho^n[z^n]g(z)$, where $g(z)$ is singular at $z = 1$. The method applies to the so-called algebraic-logarithmic functions, i.e. functions whose singular expansions involve logarithms and fractional powers. Two types of results are used. First, a catalogue of coefficients of standard functions which occur in such singular expansions, so that the coefficients of the main terms can be extracted. Second, suitable theorems which allow to extract the asymptotic order of error terms involved. Both will just be presented without proof. We refer the reader to [7, 9] for details. For our applications, only algebraic singularities can occur. Thus, the only standard functions together with the asymptotic forms of their coefficients that we will use are contained in the following table. A similar table with a lot more entries can be found in [9].

Function	Coefficient at z^n
$(1-z)^{3/2}$	$\frac{1}{\sqrt{\pi n^5}}\left(\frac{3}{4} + \frac{45}{32n} + \frac{1155}{512n^2} + \mathcal{O}(n^{-3})\right)$
$(1-z)^{1/2}$	$-\frac{1}{\sqrt{\pi n^3}}\left(\frac{1}{2} + \frac{3}{16n} + \frac{25}{256n^2} + \mathcal{O}(n^{-3})\right)$
$(1-z)^{-1/2}$	$\frac{1}{\sqrt{\pi n}}\left(1 - \frac{1}{8n} + \frac{1}{128n^2} + \frac{5}{1024n^3} + \mathcal{O}(n^{-4})\right)$
$(1-z)^{-3/2}$	$\sqrt{\frac{n}{\pi}}\left(2 + \frac{3}{4n} + \frac{7}{64n^2} + \mathcal{O}(n^{-3})\right)$

The basic requirement for the method is that the asymptotic expansion of the function should be valid in an area of the complex plane which extends beyond the disk of convergence of the original series. This requirement is described by the notions of Δ-domain and Δ-analyticity which will be introduced in the following definition.

Definition 1 ([9]) *Given two numbers ϕ, R with $R > 1$ and $0 < \phi < \frac{\pi}{2}$, the open domain $\Delta(\phi, R)$ is defined as*

$$\Delta(\phi, R) := \{z \mid |z| < R, z \neq 1, |\arg(z-1)| > \phi\}.$$

A domain is a Δ-domain if it is a $\Delta(\phi, R)$ for some R and ϕ. A function is Δ-analytic if it is analytic in some Δ-domain.

If a function f is Δ-analytic and its asymptotic expansion (including the error term) is valid for the entire Δ-domain then we are allowed to transfer f's expansion term by term into an asymptotic for f's coefficients; the error term of the expansion translates into an error term for the asymptotic. More technically we have

Theorem 1 ([9]) *Assume that $f(z)$ is Δ-analytic and that it satisfies in the intersection of a neighborhood of 1 and of its Δ-domain the condition*

$$f(z) = \circ\left((1-z)^{-\alpha}(\log\frac{1}{1-z})^{\beta}\right).$$

Then

$$[z^n]f(z) = \circ(n^{\alpha-1}(\log n)^{\beta}).$$

Here, $\circ$ is one of the operators in $\{\mathcal{O}, o\}$.

Analyticity in a Δ-domain is not a stringent requirement since the basic functions $\frac{1}{1-z}$, $\exp(z)$, $-\log(1-z)$ and $\sqrt{1-z}$ are all Δ-analytic and apart from a few degenerated exceptions the composition of these remains Δ-analytic.

We first consider $F(S, z, 1)$, i.e. the generating function $S(z, 1)$ which has as its coefficient at z^n the probability for a secondary structure of size n. There are two reasons why $S(z, 1)$ might become singular. First, the leading coefficient $c_2(z, 1)$ of $F(S, z, 1)$ might vanish as a function in z. In that case we have a reduction in the degree of F with respect to S and hence a reduction in the number of solutions. As z approaches one of these values, one observes that one or more roots of F become infinite. We thus speak of points at infinity in this case. Second, there might exist choices z_0 for z for which F has multiple roots with respect to S. In that case different roots of the equation coalesce as z approaches z_0. Note that not all of those points are necessarily singular but singularities of $S(z, 1)$ can only occur if z is one of these points. Note further that these two are the only possible reasons for a singularity of any function implicitly defined by a polynomial equation. It is well known that all these points are given by the solutions of $\mathbf{R}(F(S, z, 1), \partial_S F(S, z, 1), S) = 0$ where $\mathbf{R}$ denotes the resultant. This equation has two solutions, namely 0 and 1, where 0 counts twice. Since $S(z, 1)$ is a probability generating function, its radius of convergence is at least 1, thus 0 cannot be a singularity. Therefore $z = z_0 := 1$ is the dominant singularity in question. Note that z_0 is not a point at infinity which can be checked by setting z to z_0 within the leading coefficient of $F(S, z, 1)$. Since Newton's polygon method describes a method to find expansions at the origin, we must shift the dominant singularity to $(0, 0)$. For that purpose we determine the solution of $F(S, z_0, 1) = 0$ which is given by $S = 1$ and continue with regarding $F(Y + 1, z_0 - Z, 1)$. To this polynomial we apply Newton's polygon method. Unfortunately, it is impossible to give a precise description of this method in an extended abstract but we will show the main steps. The first step is to construct the Newton diagram for $F_1(Y, Z) := F(Y + 1, z_0 - Z, 1)$, i.e. the diagram in the (Y, Z) plane where we plot a point at (i, j) whenever the coefficient at $Y^i Z^j$ is non-zero and afterwards form the least convex polygon which contains these points. Figure 2 shows the diagram for $F_1(Y, Z)$. The polygon always has a (possibly broken) line which connects the points on the axes nearest to the origin (this is the solid line in Figure 2). The possible exponents α

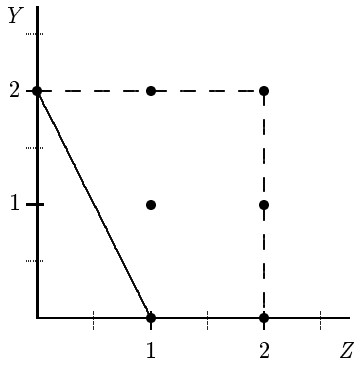


Figure 2: The Newton diagram for $F_1(Y, Z)$.

such that $Y \sim cZ^{\alpha}$ for $c \neq 0$ correspond to the negative reciprocals of the slopes of the segments of this broken line. For each such α a polynomial equation constraints the possible values of the corresponding coefficient c. For us, the only possible choice is $\alpha = \frac{1}{2}$. By solving $F_1(cZ^{1/2}, Z)$ for

c and setting $Z = 0$ afterwards (we are looking for an expansion around zero) we get $c = -2$ and thus the leading term of the expansion $-2\sqrt{Z}$. A complete expansion is obtained by repeating the process, which means deflating Y from its main term by way of the substitution $Y \mapsto Y - cZ^\alpha$. Overall we find for the desired expansion

$$-2\sqrt{Z} + 3Z - 4Z^{3/2} + \mathcal{O}(Z^2).$$

By applying the $\mathcal{O}$-transfer method (by our substitution $Z = 1 - z$ holds) we find

$$[z^n]S(z,1) \sim \left(1 + \frac{3}{8n}\right)\frac{1}{\sqrt{\pi n^3}} - \frac{3}{\sqrt{\pi n^5}},\ n \to \infty.$$

This formula gives the probability for a secondary structure of size n. The next task is to consider the number of unpaired bases, which are marked by y within $F(S,z,y)$. We therefore compute the first parital derivative of $F(S,z,y)$ with respect to y taking into consideration that S is a function in z and y (for details on that part of the method the reader is refered to [8]). Afterwards we set y to 1. Denoting $G := \partial_y S(z,y)|_{y=1}$, we find

$$\frac{1}{2}z^2SG + \frac{1}{2}zS + \frac{1}{2}zG - G$$

where S still is determined by $F(S,z,1)$. By means of resultants we can combine both polynomial equations yielding

$$F_2(G,z) := \left(4z^2 - 4z\right)G^2 + (8z - 8)\,G + z = 0$$

for the quation which determines the first parital derivative of $S(z,y)$ with respect to y at $y = 1$. The treatment of this polynomial is similar. Its dominant singulatity is located as $z = z_0 = 1$. But now, the singularity is a point at infinity and we thus substitute by $Y^2F_2(1/Y, 1-Z)$ in order to translate it into a singularity at the origin. The expansion of the resulting polynomial again is determined by Newton's polygon method. Afterwards we undo the substitution to get the following expansion

$$\frac{1}{2\sqrt{Z}} - 1 + \sqrt{Z} + \mathcal{O}(Z).$$

The application of the $\mathcal{O}$-transfer method yields

$$[z^n]\partial_y S(z,y)|_{y=1} \sim \left(\frac{1}{2} - \frac{1}{16n}\right)\frac{1}{\sqrt{\pi n}} - \frac{1}{2\sqrt{\pi n^3}},\ n \to \infty.$$

If we divide this asymptotic by that for $[z^n]S(z,1)$ we get an asymptotic for the expected number of unpaired bases in a molecule of size n. This expected value is given by $\frac{1}{2}n + \frac{3}{4} + \mathcal{O}(n^{-1})$, thus on the average there are about 50% of the bases unpaired. Note that from the mathematical point of view, the result is only valid for $n \to \infty$. However, comparing the asymptotical results with the exact coefficients of the corresponding generating functions proves that this expectation is precise up to the first two decimal places even for values of n not larger than 200. This is the usual observation when looking at asymptotics computed via the $\mathcal{O}$-transfer method. If we compute the corresponding statistic from our database we find that about 52% of the bases are unpaired; our result would have been closer to this value if we had not rounded the trained probabilities. However, our result is much stronger than a simple statistical result obtained from inspecting the database in a traditional way, since we are able to introduce the size of the molecules as a variable and prove the dependency of the parameter on the size. The assumption which is implicitly made this way is that a context-free grammar provides a realistic model for the molecules. By the

pumping property of context-free languages this assumption implies the existence of a selfsimilar behavior within the molecules which for RNA secondary structures seems to be realistic. In order to compute the variance we make use of the second factorial moment which can be computed from the second partial derivative $\partial_y^2 S(z,y)|_{y=1}$ using the same methods. We find that asymptotically the corresponding coefficient behaves like $\left(\frac{1}{4}+\frac{3}{32n}\right)\sqrt{\frac{\pi}{\pi}}-\frac{1}{4}\frac{1}{\sqrt{\pi n}}$, $n\to\infty$. Dividing this asymptotic by the asymptotic for $[z^n]S(z,1)$ and computing the variance from the first and second moment leads to $\frac{1}{4}n$ as the asymptotical representation for the variance. Thus the number of unpaired bases is not strongly conserved in the sense mentioned in the introduction.

3.2 Deriving the Results for Large Subunit RNA Molecules

As we have seen in the previous discussion, it is possible to investigate structural parameters of RNA secondary structures by small and simple context-free grammars. However, the grammar is part of the model, and different grammars, even if trained on the same database, provide slightly different results. As a consequence, in order to get consistent results, we must use one single grammar for all parameters that we want to quantify. Furthermore, it is not possible to restrict our attention to parts of a grammar only in order to investigate the structural behavior. To make this point clear, consider the stochastic context-free grammar with the following productions

$$\frac{1}{2}: S\to(S),\quad \frac{1}{2}: S\to B,\quad \frac{2}{3}: B\to|B,\quad \frac{1}{3}: B\to\varepsilon,$$

and assume that we are interested in the expected length of runs of the symbol $|$, i.e. in the expected length of subwords $|^k$ for maximal k. If we consider the entire grammar as in the previous section we find that the expected length of an $|$-run is given asymptotically by $16.48528\ldots$. One might think that it is sufficient to consider only that part of the grammar which is responsible for the generation of the $|$-runs, i.e. the two B productions. However, the generating function for this subgrammar is given by $\frac{1}{3-2z}$ and the resulting expected $|$-run length is equal to 2. As a consequence of this discussion, we must determine all parameters using the same grammar without neglecting any part of it. Thus we must use a grammar which distinguishes all substructures that we want to analyze. The following grammar $\mathcal{G}$ proved to be adequate (all capital letters are intermediate symbols):

$$f_1=S\to SAC,\ f_2=S\to C,\ f_3=C\to C|,\ f_4=C\to\varepsilon,\ f_5=A\to(L),\ f_6=L\to(L),\ f_7=L\to M$$
$$f_8=L\to I,\ f_9=L\to|H,\ f_{10}=L\to(L)B|,\ f_{11}=L\to|B(L),\ f_{12}=B\to B|,\ f_{13}=B\to\varepsilon,$$
$$f_{14}=H\to H|,\ f_{15}=H\to\varepsilon,\ f_{16}=I\to|J(L)K|,\ f_{17}=J\to J|,\ f_{18}=J\to\varepsilon,\ f_{19}=K\to K|,$$
$$f_{20}=K\to\varepsilon,\ f_{21}=M\to U(L)U(L)N,\ f_{22}=N\to U(L)N,\ f_{23}=N\to U, f_{24}=U\to U|,\ f_{25}=U\to\varepsilon.$$

The idea behind the grammar is the following: Starting at the axiom S a sentential form of the pattern $CACAC\cdots AC$ is generated, where each A stands for the starting point of a folded region and C represents a single stranded region. Applying production $A\to(L)$ produces the foundation of the folded region. From there the process has different choices. It may continue building up a ladder by applying $L\to(L)$. It may decide to introduce a multiloop by the application of $L\to M$ or an interior loop by the application of $L\to I$. A hairpin-loop is produced by $L\to|H$. Additionally, the grammar may introduce a bulge by the productions $L\to(L)B|$ resp. $L\to|B(L)$ where the two productions distinguish between a bulge at the 3' resp. 5' strand of the corresponding ladder. An interior loop is generated by the production $I\to|J(L)K|$ where J and K are used to produce the loops. The multiloop is generated by the productions $M\to U(L)U(L)N$, $N\to U(L)N$ and $N\to U$, i.e. we have at least three single stranded regions represented by U, by additional applications of the production $N\to U(L)N$ the degree of the multiloop can be increased. All

Table 2: The probabilities for the productions of our grammar obtained from training it on a database of large subunit ribosomal RNA secondary structures.

rule f	prob. p_f	rule f	prob. p_f	rule f	prob. p_f	rule f	prob. p_f	rule f	prob. p_f
f_1	0.8628	f_2	0.1372	f_3	0.9477	f_4	0.0523	f_5	1.0000
f_6	0.7612	f_7	0.0402	f_8	0.0662	f_9	0.0941	f_{10}	0.0207
f_{11}	0.0176	f_{12}	0.3730	f_{13}	0.6270	f_{14}	0.8644	f_{15}	0.1356
f_{16}	1.0000	f_{17}	0.7401	f_{18}	0.2599	f_{19}	0.7461	f_{20}	0.2539
f_{21}	1.0000	f_{22}	0.5149	f_{23}	0.4851	f_{24}	0.8137	f_{25}	0.1863

the other production-rules are used to generated unpaired regions in different contexts. We use different intermediate symbols in all cases since otherwise we would get an averaged length of the different regions instead of a distinguished length for all substructures considered.
Like for the toy-example, the next step is to adapt the probabilities for all the productions to the database. Again this is done by our Earley-parser. Table 2 presents the resulting probabilities. The system of equations which is connected with grammar $\mathcal{G}$ can be translated into a single equation using resultants or Groebner bases. Afterwards, we can proceed in exactly the same way as for the toy-example, i.e. determine the dominant singularity and compute the expansion of the generating function associated with the equation using Newton's polygon method. Afterwards this expansion is translated into an asymptotic for the coefficients of the generating function. By marking different symbols resp. the application of different productions by different variables it becomes possible to investigate numerous structural parameters of the secondary structures. For example, if we want to count the number of hairpins, we just have to mark the application of production $L \to |H$ by an additional variable like we did for symbol | for the toy-example. The results obtained this way were already presented in Table 1. Again, a comparison of our formulæ to statistics computed from the database proves that our results fit nicely with the natural behavior of the molecules. For example, the average length of a hairpin-loop observed in the database is given by 7.3748 compared to 7.3766 from Table 1. The average number of hairpins in the structures of the database is given by 52.19 which is rather close to 52.47 which we obtain by setting n to the average observed length 2321.58 within our formula. Of course these results are specific for ribosomal RNA since the database which was used to train the grammar was entirely made of LSU rRNA data. However, the methodology used is completely independent from the source of the data. Thus one can think of preparing a database from different types of RNA in order to get a model for secondary structures themselves or to build a database for other specific types of RNA such as tRNA [27] or RNASE P [2]. As already mentioned, this was done by the author. The results obtained are rather similar to those presented before, however, due to lack of space they are not reported here in detail. Furthermore, the method can be used to investigate completely different objects. For example, the author has made positive experiences in analyzing the trie data structure which is intensively used in computer science. As a consequence, it should be possible to apply the ideas presented here to other objects of biological origin, too.

4 Conclusion

In this paper we have shown how to derive expectations and higher moments related to structural parameters of RNA molecules. The model used for this purpose is rather realistic since it is derived from a database of real molecules. Furthermore, the methods used are general in the sense that they can be applied to any sort of combinatorial objects which can be modeled based on a context-free language. Thus there should be future applications in different areas such as computer science (analysis of algorithms) or bioinformatics.

References

[1] T. L. Booth, *Probability representation of formal languages*, IEEE Tenth Annual Symposium on Switching and Automata Theory, 1969.

[2] J. W. Brown, *The Ribonuclease P Database*, Nucleic Acids Res. **27** (1999), http://jwbrown.mbio.ncsu.edu/RNaseP/home.html.

[3] N. Chomsky und M. P. Schützenberger, *The Algebraic Theory of Context-Free Languages*, in Computer Programming and Formal Systems (P. Braffort and D. Hirschberg, eds.), North-Holland, Amsterdam, 1963, 118-161.

[4] R. Durbin, S. Eddy, A. Krogh and G. Mitchison, *Biological sequence analysis, Probabilistic models of proteins and nucleic acids*, Cambridge University Press.

[5] S. R. Eddy and R. Durbin, *RNA Sequence Analysis Using Covariance Models*, Nucleic Acid Res. **22** (1994), 2079-2088.

[6] H. Enomoto, T. Katayama and M. Okamoto, *Enumeration of Strings in Context-Free Languages*, Systems Computer Controls **6** (1975), 1-8.

[7] P. Flajolet and A. Odlyzko, *Singularity Analysis of Generating Functions*, SIAM J. Disc. Math. **3** (1990), No. 2, 216-240.

[8] P. Flajolet and R. Sedgewick, *The average case analysis of algorithms: Counting and generating functions*, INRIA rapport de recherche **1888**, 1993.

[9] P. Flajolet and R. Sedgewick, *The average case analysis of algorithms: complex asymptotics and generating functions*, INRIA rapport de recherche **2026**, 1993.

[10] P. Flajolet and R. Sedgewick, *Analytic Combinatorics: Functional Equations, Rational and Algebraic Functions*, INRIA rapport de recherche **4103**, 2001.

[11] W. Fontana, D. A. M. Konings, P. F. Stadler and P. Schuster, *Statistics of RNA Secondary Structures*, Biopolymers **33** (1993), 1389-1404.

[12] U. Grenander, *Syntax-Controlled Probabilities*, Tech. Rept., Division of Applied Mathematics, Brown University, 1967.

[13] E. Hille, *Analytic Function Theory*, Blaisdell Publishing Company, Waltham, 1962, 2 vol.

[14] I. L. Hofacker, P. Schuster and P. F. Stadler, *Combinatorics of RNA secondary structures*, Discrete Applied Mathematics **88** (1998), 207-237.

[15] J. E. Hopcroft, R. Motwani and J. D. Ullman, *Introduction to Automata Theorey, Languages, and Computation*, Second Edition, Addison Wesley, 2001.

[16] T. Huang and K. S. Fu, *On Stochastic Context-Free Languages*, Information Sciences **3** (1971), 201-224.

[17] S. E. Hutchins, *Moments of String and Derivation Lengths of Stochastic Context-free Grammars*, Information Sciences **4** (1972), 179-191.

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

[19] M. E. Nebel, *Combinatorial Properties of RNA Secondary Structures*, Journal of Computational Biology **9** (2002), 541-573.

[20] M. E. Nebel, *Investigation of the Bernoulli-Model of RNA secondary structures*, Frankfurter Informatik-Berichte 3/01, Institut für Informatik, Johann Wolfgang Goethe-Universität, Frankfurt a. M.

[21] R. Nussinov, G. Piecznik, J. R. Grigg and D. J. Kleitman, *Algorithms for loop matchings*, SIAM Journal on Applied Mathematics **35** (1978), 68-82.

[22] J. M. Pipas and J. E. McMahon, *Method for predicting RNA secondary structure*, Proceedings of the National Academy of Sciences **72** (1975), 2017-2021.

[23] M. Régnier, *Generating Functions in Computational Biology: a Survey*, submitted.

[24] Y. Sakakibara, M. Brown, R. Hughey, I. S. Mian, K. Sjölander, R. C. Underwood and D. Haussler, *Stochastic Context-Free Grammars for tRNA Modeling*, Nucleic Acid Res. **22** (1994), 5112-5120.

[25] D. Sankoff, *Evolution of secondary structure of 5S ribosomal RNA*, Tenth Numerical Taxonomy Conference, Kansas, 1976.

[26] D. Sankoff and J. Kruskal, *Time Warps, String Edits, and Macromolecules, The theory and Practice of Sequence Comparision*, CSLI Publications, 1999.

[27] M. Sprinzl, K. S. Vassilenko, J. Emmerich and F. Bauer, *Compilation of tRNA sequences and sequences of tRNA genes*, (20 December, 1999) http://www.uni-bayreuth.de/departments/biochemie/trna/.

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

[29] Wuyts J., De Rijk P., Van de Peer Y., Winkelmans T., De Wachter R., *The European Large Subunit Ribosomal RNA database*, Nucleic Acids Res. **29** (2001), 175-177.

[30] M. Zuker and P. Stiegler, *Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information*, Nucleic Acid Res. **9** (1981), 133-148.