Pacific Institute *for the*
Mathematical Sciences

PACIFIC INSTITUTE FOR THE MATHEMATICAL SCIENCES VIRTUAL EXPERIMENTAL MATHEMATICS LAB (PIMS VXML) FINAL REPORT: COMBINATORICS AND KNOT THEORY FOR RNA-DNA COMPLEXES

FACULTY MENTORS: MARGHERITA MARIA FERRARI[1]; CHRIS SOTEROS[2]. POSTDOC MENTOR: MATTHEW SCHMIRLER[2]. TEAM MEMBERS: JEAN LI[2]; MYKYTA SHVETS[2]; MINGZE SUN[3].

1. Introduction

DNA and RNA contain the genetic code of life. They typically occur in the form of long polymers that are subjected to high levels of confinement. During cellular processes, modifications of the geometry and topology of DNA and RNA can yield multi-stranded structures such as R-loops. An R-loop is a 3-stranded structure composed of an RNA-DNA complex and another single strand of DNA. Experimental studies indicate that R-loops can play either destructive or regulatory roles in cellular processes [1, 2, 3]. Thus, it is important to determine the factors influencing R-loop formation and stability. It is known that both DNA sequence and geometry/topology affect R-loop formation, however, little is known about their geometric and topological entanglement properties.

In order to begin to model the geometric/topological features of R-loops, in this work we develop two simplified Markov chain models of R-loops: one model is for *R-loop formation* and the other is for *R-loop geometry*.

Our simplified model for R-loop formation is inspired and informed by the work of Ferrari and coworkers [4, 5] who are developing a richer, data-informed sequence-dependent method for predicting the location of R-loops based on a formal grammar model. Given a DNA sequence, their model can be applied to predict the probabilities of R-loop formation along the DNA sequence. To complement their data-informed approach, we explore here the statistical properties of the simpler Markov

Date: Academic Year 2022–23.

1: University of Manitoba; 2: University of Saskatchewan; 3: University of Alberta.

chain model that underlies one of their formal grammar models. This is a first step towards determining to what extent, if any, randomness plays a role in R-loop formation.

Our simplified model of R-loop geometry is based on the standard statistical mechanical lattice model for polymers in a slit (2-dimensional) or tubular (3-dimensional) confinement [6, 7, 8]. We focus on a Markov chain model for the simplest 2-dimensional case and investigate how varying the transition probabilities affects the resulting R-loop geometry.

To explain our methods and results further, we first give more background about R-loops, formal grammar models for R-loops and also lattice models of polymers. Following this, for each of the models studied, we give details about the methods used and results of the study. Finally, we present conclusions and possible future directions.

2. Background and the Initial Questions

In this section, we provide background related to the project and identify the questions that motivated this work.

R-loops are 3-stranded structures formed by an RNA-DNA complex and a single strand of DNA, often appearing during transcription. Despite an increasing interest in R-loops from experimentalists, there are few mathematical studies addressing R-loop structure and formation. A first computational model for R-loop formation is presented in [9] and it is based on the sequence analysis in [10]. This work focused on identifying so-called *R-loop forming sequences* (RLFS) by analyzing certain features of a given DNA sequence related to its CG-content. This study was then expanded to develop a program for predicting and exploring RLFS [11]. More recently, R-loop formation was examined by means of a statistical mechanical equilibrium model and its software implementation called “R-looper” [12]. This energy-based model takes into account the DNA topology as well as the DNA sequence content to predict where R-loops are more likely to appear. The same group then developed a database, namely “RLBase”, to investigate R-loop datasets [13]. We refer the reader to [2, 3] for a review of the factors influencing R-loop formation and stability. In the next two subsections, we review a formal grammar model for R-loops and some background on lattice models of polymers.

2.1. Formal grammar model of R-loops overview.

A *formal grammar* is a system to generate words; it consists of a set of symbols (letters), classified as *terminals* and *non-terminals*, and a set of *production* *rules*. The production rules specify how to rewrite non-terminal symbols, so that successive applications of those rules yield words formed by only terminals. In [5], the following *regular grammar* was proposed to describe R-loop formation:

- Set of terminal symbols: $\mathcal{A} = \{\hat{\sigma}, \sigma, \hat{\tau}, \tau, \alpha, \omega\}$;
- Set of non-terminal symbols: $\mathcal{N} = \{S, Q_1, Q_2, Q_3, Q_4, Q_5, Q_6\}$ (S is called *start symbol*);
- Production rules:

$$\begin{aligned} S &\to \sigma S \mid \hat{\sigma} S \mid \sigma Q_1 \\ Q_1 &\to \alpha Q_2 \\ Q_2 &\to \hat{\tau} Q_3 \\ Q_3 &\to \tau Q_3 \mid \hat{\tau} Q_3 \mid \tau Q_4 \\ Q_4 &\to \omega Q_5 \\ Q_5 &\to \hat{\sigma} Q_6 \\ Q_6 &\to \sigma Q_6 \mid \hat{\sigma} Q_6 \mid \epsilon. \end{aligned} \tag{1}$$

Here, σ and $\hat{\sigma}$ (respectively, τ and $\hat{\tau}$) represent the length of one half-turn of DNA:DNA (respectively, RNA:DNA) complex, with the convention that *stable* interactions are denoted with a ‘ ˆ ’ (the stability of a symbol depends on the corresponding CG-content); α and ω define the initiation and termination sites of an R-loop, respectively. Note that ϵ denotes the termination of the DNA sequence. Figure 1, inspired by Figure 4 in [5], provides an example of an R-loop described by the word "...$\sigma\hat{\sigma}\sigma\sigma\alpha\hat{\tau}\tau\tau\omega\hat{\sigma}$..." from the alphabet $\mathcal{A} = \{\hat{\sigma}, \sigma, \hat{\tau}, \tau, \alpha, \omega\}$.

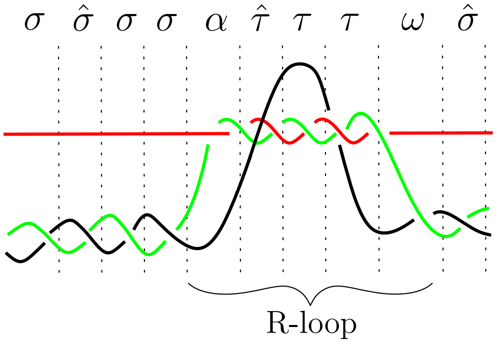


FIGURE 1. Diagram illustrating an R-loop and its corresponding word (inspired by Figure 4 in [5]). RNA-DNA complex: red/green strands, free DNA: black strand.

Since regular grammars are related to Markov chains, we utilize model 1 as the basis to explore R-loop formation and R-loop geometry in this project. In a complementary direction, Ferrari and coworkers are currently developing a probabilistic model for R-loop prediction using experimental data [4, 12]. R-loop prediction involves predicting the probability of forming an R-loop at a specific position in a given DNA sequence.

Initial Question 1: What Markov chain model underlies the formal grammar model of (1)? This is answered in Section 3.1 followed by a thorough study of the Markov chain.

2.2. **Lattice models of polymers in tubes overview.** A polymer is considered to be any large molecule that is made of repeated molecular units. Thus DNA and RNA can be thought of as polymers and indeed statistical mechanics models of polymers have proved useful for modelling the average conformational (topological/geometrical) properties of DNA in solution [14]. Such models include lattice models such as self-avoiding walk and polygon models.

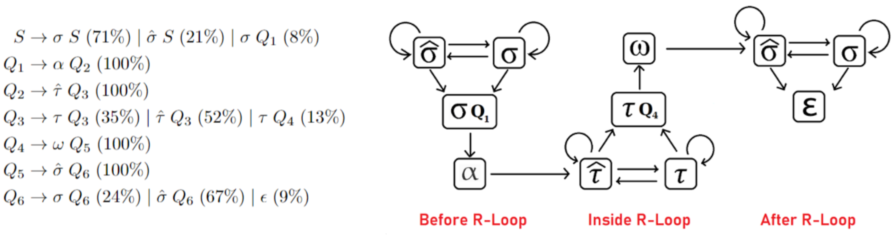
For polymers under confinement conditions, the standard lattice model considers walks or polygons confined to a tubular sublattice of the simple cubic lattice [6]. In this case, the polymer is represented by a set of vertices in $\mathbb{Z}^3$ that are joined by unit edges. For tubular confinement, the vertices are bounded in the y and z directions, with free growth allowed in the positive x-direction. For example, for the (L, M)-tube, the vertex coordinates must satisfy: $x > 0$, $0 < y \leq L$, $0 < z \leq M$. In the case of $M = 0$, the tube is 2-dimensional and called a *slit*. A mathematical advantage of lattice tube models is that they can often be studied exactly using transfer matrix and/or Markov chain methods. Soteros and coworkers [7, 8] have used such models to characterize the entanglement complexity of 2-stranded and 4-stranded polymers confined to a lattice tube.

Initial Question 2: What lattice tube model is useful for modelling the geometry/topology of R-loops? This is explored in Section 3.2 followed by a thorough study of a simple first-step model.

3. New Directions

As discussed above, the initial questions were to develop a Markov chain model associated with the formal grammar model of (1) and to develop a lattice tube model for R-loops. In order to address either of these questions, there was much to learn about both formal grammar and lattice polymer models. During the learning process, it was determined that a starting point at connecting the two approaches was to use Markov chain models and we turned our attention to two Markov chain models, one for R-loop formation and the other for R-loop geometry. The directions pursued are summarized next.

3.1. **Markov chain model of R-loop formation.** Initial Question 1 was solved first.



$S \to \sigma\ S$ (71%) | $\hat{\sigma}\ S$ (21%) | $\sigma\ Q_1$ (8%)
$Q_1 \to \alpha\ Q_2$ (100%)
$Q_2 \to \hat{\tau}\ Q_3$ (100%)
$Q_3 \to \tau\ Q_3$ (35%) | $\hat{\tau}\ Q_3$ (52%) | $\tau\ Q_4$ (13%)
$Q_4 \to \omega\ Q_5$ (100%)
$Q_5 \to \hat{\sigma}\ Q_6$ (100%)
$Q_6 \to \sigma\ Q_6$ (24%) | $\hat{\sigma}\ Q_6$ (67%) | ϵ (9%)

FIGURE 2. Left: Regular grammar from [5], with probabilities determined by using experimental data (ongoing work with Ferrari and coworkers [4]). Right: Markov chain state transition diagram corresponding to the regular grammar on the left.

To fit with the regular grammar rules, we suppose we have a discrete time Markov chain $\{X_n, n = 0, 1, 2, ...\}$ such that $X_n \in \mathcal{S} = \{\hat{\sigma}S, \sigma S, \sigma Q_1, \alpha Q_2, \hat{\tau}Q_3, \tau Q_3, \tau Q_4, \omega Q_5, \hat{\sigma}Q_6, \sigma Q_6, \epsilon\}$. A state transition diagram for the chain is shown in Figure 2 (right) (note that when the context is clear, some symbols have been shortened).Taking into account the formal grammar rules, the general one-step transition probability matrix is given by:

$$P_g =$$

$$\begin{array}{c|ccccccccccc} & \hat{\sigma}S & \sigma S & \sigma Q_1 & \alpha Q_2 & \hat{\tau}Q_3 & \tau Q_3 & \tau Q_4 & \omega Q_5 & \hat{\sigma}Q_6 & \sigma Q_6 & \epsilon \\ \hline \hat{\sigma}S & p_1 & p_2 & 1-p_1-p_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma S & p_1 & p_2 & 1-p_1-p_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma Q_1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha Q_2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{\tau}Q_3 & 0 & 0 & 0 & 0 & p_3 & p_4 & 1-p_3-p_4 & 0 & 0 & 0 & 0 \\ \tau Q_3 & 0 & 0 & 0 & 0 & p_3 & p_4 & 1-p_3-p_4 & 0 & 0 & 0 & 0 \\ \tau Q_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \omega Q_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hat{\sigma}Q_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_5 & p_6 & 1-p_5-p_6 \\ \sigma Q_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_5 & p_6 & 1-p_5-p_6 \\ \epsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} . \quad (2)$$

The entry in the ith row and jth column of P_g gives the one-step transition probability $P(X_{n+1} = j\text{th symbol}|X_n = i\text{th symbol})$ for any choice of integer $n \geq 0$. A sequence $X_0X_1...X_n$ from $\mathcal{S}$ yields a word $Y_0Y_1...Y_n$ from $\mathcal{A}$ upon deletion of all the rule symbols (capital letters). The well-known theory for finite, time-homogeneous discrete-time Markov chains [15, 16, Chapter 11] gives that the probability row vector for the distribution of X_n can be obtained by multiplying the probability row vector for the initial distribution (that of X_0) by P_g^n. From this distribution, one can determine, for example, the probability that the chain is part of an R-loop at the nth time step. Furthermore, P_g corresponds to an absorbing Markov chain where state ϵ is the lone absorbing state. The theory of finite absorbing Markov chains is well known [15, 16, Chapter 11]. In particular many other quantities of interest can be obtained from the *fundamental matrix* N_g for an absorbing Markov chain. In this case $N_g = (I - Q_g)^{-1}$ with Q_g the 10×10 submatrix of P_g obtained by deleting its last row and column. The entry in the ith row and jth column of N_g gives the expected number of time-steps that the chain visits the jth state (before being absorbed) given that it started in the ith state. Thus the expected time to absorption, after starting in the ith state, can be obtained by summing the ith row of N_g. This will give the expected total length of the word associated with a DNA sequence. To obtain the expected length of an R-loop (given starting state i for the word) one sums over the entries in the ith row of N_g that correspond to states that can occur in an R-loop: $\alpha, \hat{\tau}, \tau, \omega$.

With only 6 variables it is possible to obtain an exact solution for the fundamental matrix (for example using SageMath). The chain can also be explored by Monte Carlo computer simulation. We focussed on the choice of $p_1, ..., p_6$ as indicated in Figure 2 (left) - these were provided by Ferrari et al from analysis of the experimental data. See Section 4.1 for the exact and simulation results.

In the model of [4], letters are assigned to DNA subsequences based on experimental data [4, 12]; moreover, the length of a DNA sequence is fixed. On the other hand, for the Markov model studied here, the letters are assigned randomly according to a Markov process. Hence the Markov model generates sequences of random length and is lacking the subsequence-dependent information of the Ferrari et al model [4]. So this begs the question, why study this simplified Markov model? Towards answering this, we expanded on the Initial Question 1 as follows.

Question 1(a): What are the general properties of this simplified Markov model?
Question 1(b): Does this simplified Markov model capture any of the features of the experimental results in [12, Figure 5]?

3.2. **Markov chain model of R-loop geometry.** In order to model R-loop geometry, we focused on a model that uses polygons in the $(2,0)$-tube, a slit in the square lattice. Polygons in this slit have their left-most edges in the plane $x = 0$ and right-most edges in an integer plane $x = m$ where m is called the *span* of the polygon. See Figure 3 (left) for an example of a span 8 polygon in the slit. To model an R-loop, we use the *top walk* of the polygon (a walk from $x = 0, y = 1$ to $x = m, y = 1$ that has y-coordinates greater than or equal to 1) to represent the RNA-DNA complex of the R-loop and the remaining *bottom walk* to represent the free DNA strand. See for example Figure 3 (left) where the top walk is coloured red and the bottom walk is black.

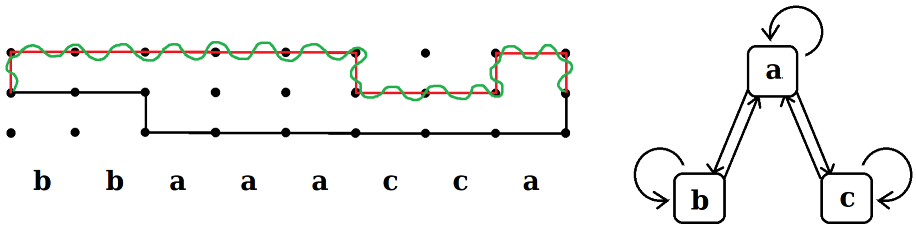


Figure 3. Left: An example of a polygon representing an R-loop embedded in the $(2,0)$-tube (slit). The top (red/green) walk represents the RNA-DNA complex and the bottom walk represents the free DNA strand. The letters underneath each half-integer plane denote the column state for that position in the R-loop polygon. Right: The state transition diagram used in the R-loop geometry Markov chain model.

Such polygons have been well studied using transfer matrix methods and we use the notation of Klein [6] to develop a Markov chain model. For this, we note that a polygon can be represented as a sequence of *column states* where in the (2,0)-tube a column state is completely defined by the location of the two edges in a particular half-integer x-plane. That is for a polygon with span m, the column state at the ith position will be defined by the location of the 2 edges in it at the plane $x = i - 1/2$. If the two edges are at height $y = 1$ and $y = 2$ then the column state is labelled b; if they are at $y = 0$ and $y = 2$, it is labelled a; and if at $y = 0$ and $y = 1$, it is labelled c. Thus a polygon can be represented by a word on the alphabet $\{a, b, c\}$. See Figure 3 (left) for the word associated with the particular polygon shown. The letters below the polygon denote the column states at positions $i = 1, ..., 8$ for this polygon with span 8.

To properly generate a polygon, the next letter in the word must correspond to a column state that can geometrically follow the previous letter. The state transition diagram in Figure 3 (right) indicates which column state transitions are allowed. We can thus consider a Markov chain $\{W_n, n = 0, 1, ...\}$ where $W_n \in \{a, b, c\}$ and the letters correspond to column states in the lattice. This chain has the general one-step transition probability matrix defined as follows:

$$P_{tube} = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} q_1 & q_2 & 1-q_1-q_2 \\ q_3 & 1-q_3 & 0 \\ q_4 & 0 & 1-q_4 \end{bmatrix} \end{array}. \tag{3}$$

The entry in the ith row and jth column of P_{tube} gives the one-step transition probability $P(W_{n+1} = j\text{th state}|W_n = i\text{th state})$ for any choice of integer $n \geq 0$. A sequence $W_0W_1...W_n$ yields a sequence of column states that form a polygon in the $(2, 0)$-tube. The well-known theory for finite, time-homogeneous discrete-time Markov chains [15, 16, Chapter 11] gives that the probability row vector for the distribution of W_n can be obtained by multiplying the probability row vector for the initial distribution (that of W_0) by P^n_{tube}. From this distribution, one can determine, for example, the probability that the chain is in a given column state at the nth time step. Furthermore, in the case that $q_i > 0, i = 1, 2, 3, 4$, the chain is a regular Markov chain and has a stationary distribution. The theory of finite regular discrete-time time-homogeneous Markov chains is well known [15, 16, Chapter 11]. In particular the *stationary distribution* and other quantities of interest can be obtained from its *fundamental matrix* Z_{tube}. In this case $Z_{tube} = (I - P_{tube} + C)^{-1}$ with C the 3×3 matrix of all ones. The probability row vector corresponding to the stationary distribution can then be obtained by multiplying a row vector of ones by Z_{tube}. Mean recurrence and mean first passage times can then be obtained using the entries of the stationary vector and Z_{tube}.

With only 4 variables, the fundamental matrix for this Markov chain can be solved exactly (for example in SageMath) as a function of $q_1, ..., q_4$.

To address Initial Question 2 for this model, the q_i's need to be chosen appropriately for modelling R-loop geometry. For this we assign the probabilities $q_1 - q_4$ in Equation 3 to take into account geometric symmetry but also to allow the top and bottom walks to have different amounts of flexibility. If a state transition results in a bending (right angle) of the bottom single stranded DNA segment, we assign it the probability p_b; note that the b in p_b stands for the bottom 'black' walk (or 'blue' used in future graphs) and is not related to the state b. Similarly, if a state transition results in a bending of the top RNA-DNA complex, we assign it the probability p_r, where r stands for the top 'red' segment. Recalling the polygon in Figure 3, we see that a transition from state b to state a results in a bending of the black walk; thus the transition is assigned the probability p_b. Similarly, a transition from state c to state a results in a bending of the red walk and thus is given the probability p_r. This yields the following transition probability matrix:

$$P_{p_r,p_b} = \begin{array}{cc} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 1-p_b-p_r & p_b & p_r \\ p_b & 1-p_b & 0 \\ p_r & 0 & 1-p_r \end{bmatrix} \end{array}. \tag{4}$$

Since the top walk represents a double helix, it is expected to be less flexible than the bottom walk representing a single DNA strand. Hence we assume $p_r < p_b$. In particular, more flexibility is expected to mean more right angles (bends) on average. This leads to a refinement of Initial Question 2.

Question 2(a): What are the geometric properties of the generated R-loops as a function of p_r, p_b?

4. Progress

In this section, we discuss our computational work to address Questions 1(a),1(b), and 2(a).

4.1. **Progress on the Markov chain model of R-loop formation.** Significant progress has been made in our ongoing research on modeling R-loop formation using Markov chains. As part of our efforts we developed several computer programs:

- `M. Shvets' Python Code 1`: This code, written in Python 3, generates words from the alphabet $\mathcal{A}$ using a Monte Carlo simulation approach. The code starts at a given state and chooses the next state randomly based on the current state and the one-step transition probabilities. The chain progresses step by step and the user can advance to the next step by pressing enter, allowing for easy visualization of the generated words. The code also includes the option to generate a series of random words and calculate the average length of the R-loops. See Figure 4 (left) for sample output. Furthermore, it generates a frequency table of the start positions of the R-loops, which is saved to a csv file for further analysis. The code is modular and can be easily modified to generate different words and frequency tables.

- `M. Shvets' Scala Code 1`: This code, written in Scala, focuses on efficiently collecting statistical data during the simulation of R-loop formation. It incorporates tracker functions that are called after each state transition to keep track of R-loops and their lengths. These tracker functions can be combined, allowing for multiple tracker functions to be used together. The code also includes a main function that runs the simulation for a given number of iterations and saves the results to a csv file, which can be easily analyzed and plotted to generate graphs.
- `M. Sun's Python Code 1`: This code, written in Python 3, also uses the Markov chain defined in (2) to simulate the generation of words from the alphabet $\mathcal{A}$. In addition it is designed to pick out words with a pre-defined length. The final outputs are the average length of an R-loop and a graph showing the probability of being part of an R-loop at each time step within a fixed length word.
- `J. Li's SageMath Code 1`: This code, written in SageMath Version 9.7 in CoCalc, solves for the fundamental matrix of the absorbing Markov chain as a function of $p_1, ..., p_6$. The results are then used to obtain exact values for the expected total word length, the expected length of an R-loop, and the position-dependent probability of being in an R-loop.

The results regarding Question 1(a) are as follows. We focused on the specific choice of p_i's from Figure 2 (left): $p_1 = 0.21, p_2 = 0.71, p_3 = 0.52, p_4 = 0.35, p_5 = 0.67, p_6 = 0.24$. We explored the average word length and the average R-loop length exactly, via the SageMath program, and statistically, via the Monte Carlo simulation programs. States of an R-loop are considered to be: $\{\alpha, \tau, \hat{\tau}, \omega\}$. The results from both approaches were consistent with each other. For example, the following were obtained for the case that $X_0 = \hat{\sigma} S$.

Average Word length (including first ϵ):

Simulations: 600 time-steps, 10000 replications: 36.296 +/- 0.606 letters;

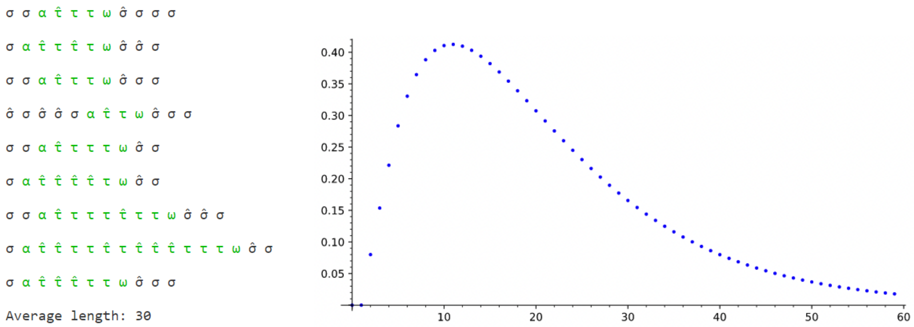
Exact: 36.3034188034188 letters.

Average R-loop length:

Simulations: 600 time-steps, 1000 replications: 10.568 +/- 0.461 letters;

Exact: 10.6923076923077 letters.

We also calculated the probability of a position in a word being part of an R-loop (i.e. in one of the states $\{\alpha, \tau, \hat{\tau}, \omega\}$). Figure 4 (right) shows the exact results from the SageMath program for the case $X_0 = \hat{\sigma}S$. The Monte Carlo results (not shown) are consistent.



```
σ σ α τ̂ τ τ ω σ̂ σ σ σ
σ α τ̂ τ τ̂ τ ω σ̂ σ̂ σ
σ σ α τ̂ τ τ ω σ̂ σ σ
σ̂ σ σ̂ σ̂ σ α τ̂ τ ω σ̂ σ σ
σ σ α τ̂ τ τ τ ω σ̂ σ
σ α τ̂ τ̂ τ̂ τ̂ τ ω σ̂ σ
σ σ α τ̂ τ τ τ τ̂ τ τ ω σ̂ σ̂ σ
σ α τ̂ τ̂ τ τ τ̂ τ τ̂ τ̂ τ̂ τ τ τ ω σ̂ σ
σ α τ̂ τ̂ τ̂ τ τ ω σ̂ σ σ
Average length: 30
```

Figure 4. Left: Sample output words from M. Shvets' Python Code 1 with R-loops colored in green. Right: The probability of the chain being in an R-loop state at time-step n, i.e. $P(X_n \in \{\alpha, \tau, \hat{\tau}, \omega\} | X_0 = \hat{\sigma}S), n = 0, 1, 2, ...$ generated from J. Li's SageMath Code 1.

Regarding Question 1(b), we consider whether we can make any connections between the Markov model results and those for DNA. The probabilities used were based on an analysis of data for DNA sequences of length 1500-1800 nucleotides and letters were assumed to be associated with subsequences 4 nucleotides in length [4, 12]. For one experiment on such a DNA sequence, the average length of an R-loop was about 130 nucleotides [12, Figure 5 (C) (top)]. So R-loop length is roughly one tenth DNA length while from our random model it is closer to one third. [12, Figure 5 (A) (bottom)] shows experimental (red) probabilities of being in an R-loop as a function of DNA sequence position. For this experiment, the probability starts out very low and only begins to rise near the 500th-600th nucleotide, then reaches a maximum of about 0.5 around the 700th nucleotide (near half way along the DNA sequence) and then drops to very low again around the

800th-850th nucleotide. While for our Markov model the maximum is close to 0.4 and is achieved around the 10th letter, which is only about one third of the way along an average word.

One major problem with the Markov model is that the word length is allowed to vary. A better way to compare the model to experimental data would be to focus only on words of length 375 to correspond to a DNA sequence of length 1500. However, based on the exact calculations, the probability of generating a word of length 375 from the Markov model is on the order of 10^{-11}. Calculating conditional probabilities exactly for word length 375 seemed to be problematic and Monte Carlo generation was not possible given the time limitations of the project. As such, to explore the effect of fixing the word length on the model properties, we focused on generating words of length 100 from the model - the probability of such words being on the order of 10^{-4}. Figure 5 shows the results from the Python simulation for the position-dependent probability of being part of an R-loop. We see that the distribution is much broader and the maximum is slightly lower than for the unconditioned case. The location of the peak is slightly greater than half way along the word. Based on this, it seems unlikely that just conditioning on words of a fixed length will capture all the features of the experimental data. Thus, not unexpectedly, more information about the DNA sequence needs to be incorporated into the model.

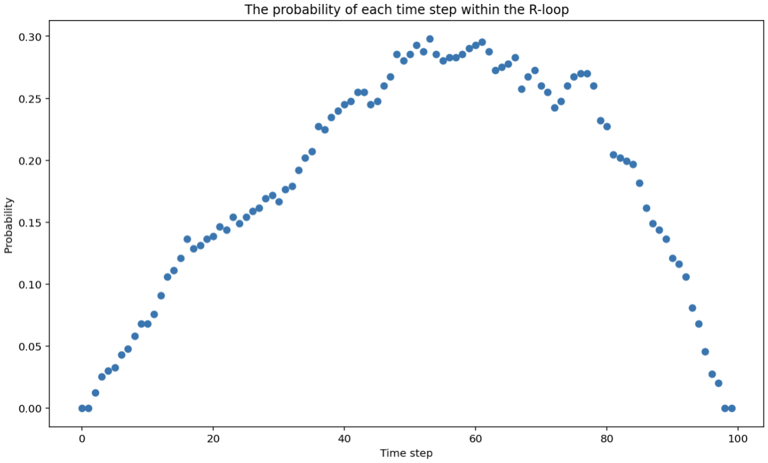
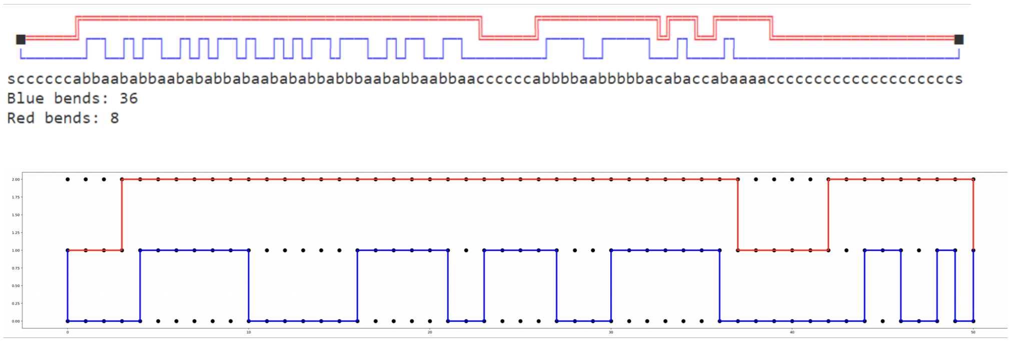


Figure 5. The probability of being in an R-loop as a function of the position (time-step) in a word of length 100. Results were generated from M. Sun's Python Code 1 with 10^6 replications.

4.2. **Progress on the Markov chain model of R-loop geometry.** We worked on developing a Markov chain model to explore the geometry of R-loops using a lattice tube model.

- `M. Shvets' Python Code 2`: The code, written in Python 3, implements a simulation of the Markov chain of column states within the lattice tube using a Monte Carlo approach. The probabilities of transitioning to different states in the lattice tube are determined by the probabilities of red (top walk) and blue (bottom walk) bends, denoted as p_r and p_b respectively. The red strand represents the RNA-DNA complex and is less flexible, while the blue strand represents DNA and is more flexible. See Figure 6 (top) for sample output. The code is modular, making it easy to customize various parameters such as the probabilities of transitioning to different states, the length of the chain, and the number of chains to generate. The chain is visualized both as a console output and as a diagram using Unicode characters, providing a clear representation of the chain's geometry. Furthermore, the code allows for counting the bends in the chain and recording the data, which can be plotted as a histogram using libraries such as Matplotlib and NumPy. This enables us to analyze the distribution of bends in the R-loop geometry and gain insights into the overall structure.
- `M. Sun's Python Code 2`: This code, written in Python 3, also simulates the Markov chain of column states in the $(2,0)$-tube. The output graph shows the coordinates of the two strands and also visualizes the sequence of column states. See Figure 6 (bottom) for sample output.

Regarding Question 2(a), Figure 7 illustrates the conclusions of our study so far. Namely, as expected, favouring bends in the bottom walk over bends in the top walk leads to the majority of bends being in the bottom walk. We also note that the stationary distribution for the Markov chain has been solved exactly and the Monte Carlo results for it are consistent. Obtaining the exact bend distribution is also possible but this work has yet to be completed.



```
sccccccabbaababbaababababbabaababababbabbbaababbaabbaaccccccabbbbaabbbbbacabaccabaaaaccccccccccccccccccccs
Blue bends: 36
Red bends: 8
```

Figure 6. Top: Sample output of M. Shvets' Python Code 2 for span 100. Bottom: Sample output of M. Sun's Python Code 2 for span 50. In both graphs, the RNA-DNA complex is in red and the free DNA is in blue; bending parameters are $p_r = 0.1$, $p_b = 0.5$.

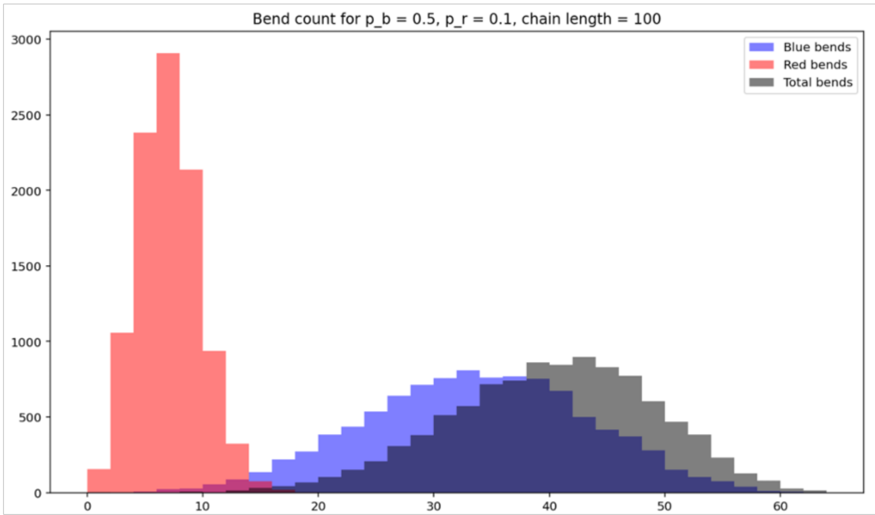


Figure 7. Histograms of bend counts in R-loops for the RNA-DNA complex (red), free DNA (blue), and total bends (grey).

5. Conclusions and Future Directions

In summary, two Markov models have been studied related to modelling R-loops. Exact and Monte Carlo analyses have been performed for each. The results from both approaches are consistent. For the model of R-loop formation, we conclude that a purely random model is not sufficient to capture the experimental data about R-loop formation in DNA. For the model of R-loop geometry, we found even a simplified model can take into account differing flexibility for the components of an R-loop.

Future directions will involve expanding the study of the R-loop geometry; for example, by visualizing lattice tube configurations in the 3-dimensional space via KnotPlot [17]. This will require understanding how to assign a letter from the alphabet $\mathcal{A} = \{\hat{\sigma}, \sigma, \hat{\tau}, \tau, \alpha, \omega\}$ to geometric configurations, as well as incorporating experimental information into the model.

Acknowledgments

The team acknowledges helpful discussions with M. Vazquez, J. Lusk, R. Scharein and P. Pongtanapaisan. MMF acknowledges helpful discussions with N. Jonoska, S. Poznanović, M. Riehl and the Chédin Lab. We are also grateful for the CoCalc accounts provided by PIMS and Overleaf accounts provided by MMF and CS. CS acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference number RGPIN-2020-06339].

References

[1] F. Chédin. "Nascent connections: R-loops and chromatin patterning". *Trends in Genetics* **32**.12 (2016), pp. 828–838.

[2] F. Chédin and C. J. Benham. "Emerging roles for R-loop structures in the management of topological stress". *Journal of Biological Chemistry* **295**.14 (2020), pp. 4684–4695.

[3] Y. A. Hegazy, C. M. Fernando, and E. J. Tran. "The balancing act of R-loop biology: The good, the bad, and the ugly". *Journal of Biological Chemistry* **295**.4 (2020), pp. 905–913.

[4] M. M. Ferrari, S. Poznanović, M. Riehl, J. Lusk, S. Hartono, F. Chédin, M. Vazquez, and N. Jonoska. "A data driven method for R-loop prediction". *In preparation* (2023).

[5] N. Jonoska, N. Obatake, S. Poznanović, C. Price, M. Riehl, and M. Vazquez. "Modeling RNA:DNA Hybrids with Formal Grammars". *Using Mathematics to Understand Biological Complexity: From Cells to Populations*. Ed. by R. Segal, B. Shtylla, and S. Sindi. Cham: Springer International Publishing, 2021, pp. 35–54. DOI: 10.1007/978-3-030-57129-0_3.

[6] D. J. Klein. "Asymptotic distributions for self-avoiding walks constrained to strips, cylinders, and tubes". *Journal of Statistical Physics* **23**.5 (1980).

[7] N. R. Beaton, J. W. Eng, and C. E. Soteros. "Knotting statistics for polygons in lattice tubes". *Journal of Physics A: Mathematical and Theoretical* **52**.14 (2019).

[8] J. W. Eng. “A transfer matrix approach to studying the entanglement complexity of self-avoiding polygons in lattice tubes”. PhD thesis. University of Saskatchewan, 2020.

[9] T. Wongsurawat, P. Jenjaroenpun, C. K. Kwoh, and V. Kuznetsov. “Quantitative model of R-loop forming structures reveals a novel level of RNA–DNA interactome complexity”. *Nucleic Acids Research* **40**.2 (2012), e16–e16.

[10] D. Roy and M. R. Lieber. “G clustering is important for the initiation of transcription-induced R-loops in vitro, whereas high G density without clustering is sufficient thereafter”. *Molecular and Cellular Biology* **29**.11 (2009), pp. 3124–3133.

[11] P. Jenjaroenpun, T. Wongsurawat, S. P. Yenamandra, and V. A. Kuznetsov. “QmRLFS-finder: a model, web server and stand-alone tool for prediction and analysis of R-loop forming sequences”. *Nucleic Acids Research* **43**.W1 (2015), W527–W534.

[12] R. Stolz, S. Sulthana, S. R. Hartono, M. Malig, C. J. Benham, and F. Chédin. “Interplay between DNA sequence and negative superhelicity drives R-loop structures”. *Proceedings of the National Academy of Sciences* **116**.13 (2019), pp. 6260–6269.

[13] H. E. Miller et al. “Exploration and analysis of R-loop mapping data with RLBase”. *Nucleic Acids Research* **51**.D1 (2023), pp. D1129–D1137.

[14] E. Orlandini and S. G. Whittington. “Statistical topology of closed curves: Some applications in polymer physics”. *Reviews of Modern Physics* **79** (2007).

[15] J. Snell. *Introduction to Probability.* Birkhauser Mathematics Series. Random House, 1988.

[16] C. M. Grinstead and J. L. Snell. *Introduction to Probability.* AMS, 2006.

[17] R. Scharein. “The KnotPlot Site”. http://knotplot.com.