StreAM-T_g : Algorithms for Analyzing Coarse Grained RNA Dynamics Based on Markov Models of Connectivity-Graphs

Sven Jager[1], Benjamin Schiller[2], Thorsten Strufe[2], and Kay Hamacher[1,3]

[1] Computational Biology and Simulation, Dept. of Biology, TU Darmstadt, Darmstadt, Germany {jager,hamacher}@bio.tu-darmstadt.de
[2] Privacy and Data Security, Dept. of Computer Science, TU Dresden, Dresden, Germany {benjamin.schiller1,thorsten.strufe}@tu-dresden.de
[3] Dept. of Physics, Dept. of Computer Science, TU Darmstadt, Darmstadt, Germany

Abstract. In this work, we present a new coarse grained representation of RNA dynamics. It is based on cliques and their patterns within adjacency matrices obtained from molecular dynamics simulations. RNA molecules are well-suited for this representation due to their composition which is mainly modular and assessable by the secondary structure alone. Each adjacency matrix represents the interactions of *k* nucleotides. We then define transitions between states as changes in the adjacency matrices which form a Markovian dynamics. The intense computational demand for deriving the transition probability matrices prompted us to develop *StreAM-T_g*, a stream-based algorithm for generating such Markov models of *k*-vertex adjacency matrices representing the RNA. Here, we benchmark *StreAM-T_g* a) for random and RNA unit sphere dynamic graphs. b) we apply our method on a long term molecular dynamics simulation of a synthetic riboswitch (1,000 ns). In the light of experimental data our results show important design opportunities for the riboswitch.

Keywords: RNA, Markovian Dynamics, Dynamic Graphs, Molecular Dynamics, Coarse Graining, Synthetic Biology

1 Introduction

The computational design of switchable and catalytic ribonucleic acids (RNA) becomes a major challenge for synthetic biology [23]. So far, available models and simulation tools to design and analyze functionally complex RNA based devices are very limited [8]. Although several tools are available to assess secondary as well as tertiary RNA structure [3], current capabilities to simulate dynamics are still underdeveloped [16] and rely heavily on atomistic molecular dynamics (MD) techniques [14]. RNA structure is largely modular and composed of repetitive motifs [16] that form structural elements such as hairpins and stems based on hydrogen- bonding patterns [30]. Such structural modules play an important role for nano design [23,19].

In order to understand RNA dynamics [1,26] we develop a new method to quantify all possible structural transitions, based on a coarse grained, transferable representation of different module sizes. The computation of Markov State Models have recently become practical to reproduce long-time conformational dynamics of biomolecules using data from MD simulations [9]. To this end, we convert MD trajectories into dynamic graphs and derive the Markovian dynamics in the space of adjacency matrices. Aggregated matrices for each nucleotide represent RNA coarse grained dynamics. However, a full computation is computationally expensive.

To address this challenge we extend *Stream* - a stream based algorithm for counting motifs in dynamic graphs with an outstanding performance of counting motifs in biomolecular trajectories [22]. The extension *StreAM* computes one transition matrix for a single set of vertices or a full set for combinatorial many matrices. To gain insight into global folding and stability, we propose *StreAM-T_g*: It combines all Markov models for an RNA into a global weighted stochastic transition matrix T_g.

The remainder of this paper is structured as follows: In Sec. 2, we introduce the concept as well as our biological test setup. We describe details in Sec. 3. We present run-time evaluations of our algorithm in Sec. 4 for a synthetic tetracycline (TC) dependent Riboswitch (TC-Aptamer). Finally, we summarize our work in Sec. 5.

2 Our Approach for Coarse Grained Analysis

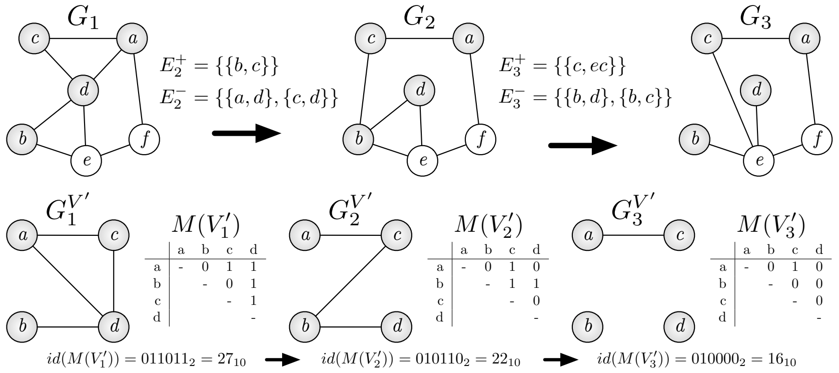
2.1 Structural Representation of RNA

Predicting the function of complex RNA molecules depends critically on understanding both, their structure as well as their conformational dynamics [15,18]. To achieve the latter we propose a new coarse grained RNA representation and the dynamics in the implied state space at the nucleotide level. For our approach, we start with a MD simulation to obtain a trajectory of the RNA. We reduce these simulated trajectories to nucleotides represented by their ($C3'$) atoms. From there, we represent RNA structure as an undirected graph [11] using each $C3'$ as a vertex and distance dependent interactions as edges [3]. It is well known that nucleotide-based molecular interactions take place between more than one partner [28]. For this reason interactions exist for several edges observable in the adjacency matrix (obtained via a Euclidean distance cut-off) of $C3'$ coordinates at a given time-step. The resulting edges represent, e.g., strong local interactions such as Watson-Crick pairing, Hoogsteen, or $\pi-\pi$-stacking.

Our algorithm estimates adjacency matrix transition rates of a given set of vertices (nucleotides) and builds a Markov model. Moreover, by deriving all Markov models of all possible combinations of vertices, we can reduce them afterwards into a global weighted transition matrix for each vertex representing the ensemble that the vertex/nucleotide is immersed in.

2.2 Dynamic Graphs, their Analysis and Markovian Dynamics

A *graph* $G=(V,E)$ is an ordered pair of *vertices* $V=\{v_1,v_2,\dots v_{|V|}\}$ and *edges* E. Here, we only consider *undirected graphs without loops*, i.e., $E\subseteq\{\{v,w\}:v,w\in V,v\neq w\}$. For a subset V' of the vertex set V, we refer to $G^{V'}=(V',E'),E':=\{\{v,w\}\in E:v,w\in V'\}$ as the V'*-induced subgraph* of G.


G_1
$E_2^+ = \{\{b,c\}\}$
$E_2^- = \{\{a,d\},\{c,d\}\}$
G_2
$E_3^+ = \{\{c,ec\}\}$
$E_3^- = \{\{b,d\},\{b,c\}\}$
G_3
a
b
c
d
e
f
$G_1^{V'}$
$G_2^{V'}$
$G_3^{V'}$
$id(M(V_1')) = 011011_2 = 27_{10}$
$id(M(V_2')) = 010110_2 = 22_{10}$
$id(M(V_3')) = 010000_2 = 16_{10}$

$M(V_1')$

	a	b	c	d
a	-	0	1	1
b		-	0	1
c			-	1
d				-

$M(V_2')$

	a	b	c	d
a	-	0	1	0
b		-	1	1
c			-	0
d				-

$M(V_3')$

	a	b	c	d
a	-	0	1	0
b		-	0	0
c			-	0
d				-

Fig. 1: Example of a dynamic graph and induced subgraphs for $V' = (a,b,c,d)$

The *adjacency matrix* $A(G) = A_{i,j}$ of a graph G is a $|V| \times |V|$ matrix, defined as follows:

$$A_{i,j} := \begin{cases} 0 : i < j \wedge \{v_i, v_j\} \notin E \\ 1 : i < j \wedge \{v_i, v_j\} \in E \\ \uparrow : \text{otherwise} \end{cases} \tag{1}$$

We denote the set of all adjacency matrices of size k as $\mathscr{A}_k$, with $|\mathscr{A}_k| = 2^{\frac{k\cdot(k-1)}{2}}$. With $concat(A)$, we denote the row-by-row *concatenation* of all defined values of an adjacency matrix A. We define the *adjacency id* of a matrix A as the numerical value of the binary interpretation of its concatenation, i.e., $id(A) = concat(A)_2 \in \mathbb{N}$. We refer to $id(V') := id(A(G^{V'}))$ as the adjacency id of the V'-induced subgraph of G. For example, the concatenation of the adjacency matrix of graph $G_1^{V'}$ (shown in Fig. 1) is $concat(A(G_1^{V'})) = 011011$ and its adjacency id is $id(V') = 011011_2 = 27_{10}$.

As a *dynamic graph* $G_t = (V, E_t)$, we consider a graph whose edge set changes over time. For each point in time $t \in [1, \tau]$, we consider G_t as the *snapshot* or *state* of the dynamic graph at that time. The *transition of a dynamic graph* G_{t-1} to the next state G_t is described by a pair of edge sets which contain the edges added to and removed from G_{t-1}, i.e., (E_t^+, E_t^-). We refer to these changes as a *batch*, defined as follows: $E_t^+ := E_t \backslash E_{t-1}$ and $E_t^- := E_{t-1} \backslash E_t$. The *batch size* is referred as $\delta_t = |E_t^+| + |E_t^-|$.

The *analysis* of dynamic graphs is commonly performed using *stream-* or *batch-based* algorithms. Both output the desired result for each snapshot G_t. Stream-based algorithms take a single update to the graph as input, i.e., the addition or removal of an edge e. Batch-based algorithms take a pair (E_{t+1}^+, E_{t+1}^-) as input. They can always be implemented by executing a stream-based algorithm for each edge addition $e \in E_{t+1}^+$ and removal $e \in E_{t+1}^-$.

The result of analyzing the adjacency id of V' for a dynamic graph G_t is a list $(id_t(V') : t \in [1, \tau])$. We consider each pair $(id_t(V'), id_{t+1}(V'))$ as an *adjacency transition of* V' and denote the *set of all transitions* as $\mathscr{T}(V')$. Then, we define the *local transition matrix* $T_{i,j}(V')$ of V' as a $|\mathscr{A}_k| \times |\mathscr{A}_k|$ matrix which contains the number of transitions between any two adjacency ids over time, i.e., $T_{i,j}(V') := |(i+1, j+1) \in \mathscr{T}(V')|$. From $T_{i,j}(V')$, we can derive a *Markov model* to describe these transitions.

By combining all possible $T_{i,j}(V')$, a specific vertex v is immersed in a subset V', we derive a transition tensor $C_{i,j,l}(V')$ with dimensions $|\mathscr{A}_k| \times |\mathscr{A}_k| \times (k-1)!\binom{|V|}{k-1}$. We define a global weighting parameter w_l, by considering the local distribution weighted by its global distribution of transitions matrices. A global transition matrix T_g is defined as $\sum_l w_l \times C_l(V')$ with the dimensions $|\mathscr{A}_k| \times |\mathscr{A}_k|$.

For a local or global transition matrix the respective dominant eigenvector[4] is called π and represents the stationary distribution attained for infinite (or very long) times. The corresponding conformational entropy of the ensemble of motifs is $H := -\sum_i \pi_i \cdot \log \pi_i$. The change in conformational entropy upon, e.g., binding a ligand is then given as $\Delta H = H_{wt} - H_{complex}$.

2.3 Workflow

MD Simulation Setup. We use a structure of a synthetic tetracycline binding riboswitch (PDB: 3EGZ, chain B, resolution: 2.2 Å, Figure 2) [32] and perform two simulations: the riboswitch with tetracycline in complex and without tetracycline. As tetracycline binding alters the structural entropy of the molecule [31] our proposed method should be able to detect changes in (local) dynamics due the presence of tetracycline. Both simulations were performed with Gromacs using the charmm27 force field with parameters for synthetic tetracycline derivatives (7CL) [2,5,20]. Simulations were performed at constant temperature (300K) and pressure (1 bar). The simulation box is filled with Tip3p water, and Mg^{2+} counter ions. After minimization we equilibrate the solvent with fixed RNA for 10 ns, release the RNA and started the simulation with an integration step-size of 1.5 fs. Both all-atom MD simulations last for 1,000 ns.

Graph Transformation. Both MD simulations contain 160,000 snapshots. We generated dynamic graphs $G_t = (V, E_t)$ containing $|V| = 65$ vertices (Tab. 1), each modelling a nucleic $3C'$ (Fig. 2). This resolution is sufficient to represent both small secondary structure elements as well as large quaternary RNA complexes [10,12]. We create undirected edges between two vertices in case their Euclidean cut-off (d) is shorter than $d \in [10, 15]$Å (cmp. Tab. 1).

Markov State Models (MSM) of Local Adjacency Transitions. *StreAM* counts transitions of a k-vertex set with a size of $|\mathscr{A}_k|$ for a given set of V' obtained from a dynamic graph $G_t(V, E_t)$. Afterwards, we can compute respective probabilities resulting in a transition matrix: $T_{i,j}(V') = |(i+1, j+1) \in \mathscr{T}(V')|$. Not all possible states are necessarily visited in a given, finite simulation, although a “missing state” potentially might occur in longer simulations. In order to allow for this, we introduce a minimal pseudo-count [25] of $P_k = \frac{1}{|\mathscr{A}_k|}$.

[4] guaranteed to exist due to the Perron-Frobenius theorem with an eigenvalue of $\lambda = 1$

Global Transition Matrix. Here $C_{i,j,l}(V)$ is the count tensor of transitions between i and j in matrix $\mathscr{T}(V')$. It contains all $T_{i,j}(V')$ a specific vertex is immersed in and due to this it contains all possible information of local markovian dynamcis. $C_{i,j,l}(V)$ is normalized by the count of all transitions of i in all matrices $S_{j,l} = \sum_i C_{i,j,l}(V)$. The global weighting parameter $w_l = \frac{S_{jl}}{\sum_l S_{jl}}$ can be derived by taking all transitions $\mathscr{A}_k$ into account with respect to their probability. For a given set of l transition matrices $T_{i,j}(V')$ we can combine them into a global model:

$$T_{gi,j}(V) = \sum_l \frac{S_{jl}}{\sum_l S_{jl}} \cdot C_{i,j,l}(V). \tag{2}$$

Stationary Distribution and Entropy. As T_g is a row stochastic matrix we can compute its dominant eigenvector from a spectral decomposition. It represents a basic quantity of interest: the stationary probability $\pi := (\pi_1, \ldots, \pi_i, \ldots)$ of micro-states i [25]. To this end we used the `markovchain` library in `R` [27,29]. For measuring the changes in conformational entropy $H := -\sum_{i=1}^{|\mathscr{A}_k|} \pi_i \cdot \log \pi_i$ upon binding a ligand, we define $\Delta H = H_{wt} - H_{complex}$, form a stationary distribution.

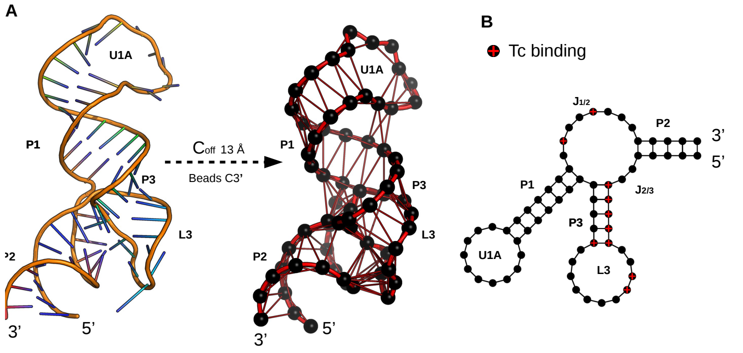


Fig. 2: Structural representation of TC-Aptamer. **A:** TC-Aptamer with a cut-off of 13Å and using $C3'$ atom for coarse graining reveals edges for dominant WC base-pairings. **B:** Secondary structure representation of TC-Aptamer. Nucleotides participating in TC binding are colored in red. Graphics were created using `Pymol` and `R` [24,29].

3 Algorithm

***StreAM* and *StreAM$_B$*.** We compute the adjacency id $id(V')$ for vertices $V' \subseteq V$ in the dynamic graph G_t using the stream-based algorithm *StreAM*, as described in Algorithm 1. Here, $id(V') \in [0, |\mathscr{A}_{|V'|})$ is the unique identifier of the adjacency matrix of the subgraph $G^{V'}$. Each change to G_t consists of the edge $\{a,b\}$ and a type to mark it as addition or removal (abbreviated to *add*,*rem*). In addition to edge and type, *StreAM* takes as input the ordered list of vertices V' and their current adjacency id.

An edge $\{a,b\}$ is only processed by *StreAM* in case both a and b are contained in V'. Otherwise, its addition or removal has clearly no impact on $id(V')$.

Assume $pos(V',a), pos(V',b) \in [1,k]$ to be the positions of vertices a and b in V'. Then, $i = min(pos(V',a), pos(V',b))$ and $j = max(pos(V',a), pos(V',b))$ are the row and column of adjacency matrix $A(G^{V'})$ that represent the edge $\{a,b\}$. In the bit representation of its adjacency id $id(V')$, this edge is represented by the bit $(i-1)\cdot k+j-i\cdot(i+1)/2$. When interpreting this bit representation as a number, an addition or removal of the respective edge corresponds to the addition or subtraction of $2^{k\cdot(k-1)/2-((i-1)\cdot k+j-i\cdot(i+1)/2)}$. This operation is performed to update $id(V')$ for each edge removal or addition. In the following, we refer to this position as $e(a,b,V') := \frac{|V'|\cdot(|V'|-1)}{2} - ((i-1)\cdot|V'|+j-\frac{i\cdot(i+1)}{2})$.

```
Data: $V'$, $id$, $\{a,b\}$, $type \in \{add, rem\}$
begin
    if $a \in V' \wedge b \in V'$ ;                /* process only relevant edges */
     then
        if $type == add$ then
            $A := A + 2^{e(a,b,V')}$ ;          /* set corresponding bit to 1 */
        else
            $A := A - 2^{e(a,b,V')}$ ;          /* set corresponding bit to 0 */
        end
    end
    return $id$ ;
end
```

Algorithm 1: *StreAM*: stream-based computation of the adjacency id

Furthermore, in Algo. 2 we show $StreAM_B$ for the batch-based computation of the adjacency id for vertices V'

```
Data: $V'$, $id_{t-1}$, $E_t^+$, $E_t^-$
begin
    $id_t(V') := id_{t-1}(V')$ ;                  /* init id with previous one */
    for all $\{a,b\} \in E_t^+$ do
        $id_t := StreAM(V', id_t, \{a,b\}, add)$ ;         /* process addition */
    end
    for all $\{a,b\} \in E_t^-$ do
        $id_t := StreAM(V', id_t, \{a,b\}, rem)$ ;          /* process removal */
    end
    return $id_t$ ;
end
```

Algorithm 2: $StreAM_B$: batch-based computation of the adjacency id

***StreAM-T_g*.** We present *StreAM-T_g*, an algorithm for the computation of global transition matrices, one particular vertex is participating in, given in Algorithm 3. A full computation with *StreAM-T_g* can be divided into the following steps. The first step is the computation of all possible Markov models with *StreAM* from all $\binom{|V|}{k} \cdot k! = \frac{|V|!}{(|V|-k)!}$ combinations, where k is the adjacency size and $|V|$ the number of vertices of G_t. Afterwards, *StreAM-T_g* sorts the matrices by vertex id into different sets, each with the size of $\binom{|V|}{k-1} \cdot (k-1)!$. For each vertex, *StreAM-T_g* computes a global count tensor C which is normalized by the global distribution of transition states a vertex is immersing in, taking the whole ensemble into account.

```
Data: $T, a$
begin
    $C(V) := \{V' \in \mathbb{P}(V) : |V'| = k, a \in V'\}$ ;    /* C vertex $a$ immersed in */
    $T_g(a) := 0_{|\mathscr{A}_k|,|\mathscr{A}_k|}$ ;    /* initialize $T_g(a)$ */
    for all $V' \in C(V)$ do
        $T_g(a) := T_g(a) + \frac{S(V')}{\sum_{V'' \in C(V)} |S(V'')|} \cdot T(V')$ ;    /* sum up $T_g(a)$ */
    end
    return $T_g(a)$
end
```

Algorithm 3: *StreAM-$T_g(a)$* for computing the global transition matrix $T_g(a)$

4 Evaluation

4.1 Objectives

As *StreAM-T_g* is intended to analyze large MD trajectories we first measured the speed of *StreAM* for computing a single $\mathscr{T}(V')$ to estimate overall computational resources. With this in mind, we benchmark different G_t with increasing adjacency size k (Tab. 1). Furthermore, we need to quantify the dependence of computational speed with respect to δ_t. Note, δ_t represents changes in conformations within G_t. For the full computation of T_g, we want to measure computing time in order to benchmark *StreAM-T_g* by increasing network size $|V|$ and k for a given system due to exponentially increasing matrix dimesnions $|\mathscr{A}_k| = 2^{\frac{k \cdot (k-1)}{2}}$. We expect due to combinatorial complexity of matrix computation a linear relation between $|V|$ and speed and an exponential relation between increasing k and speed. For the last part, we want to compare Markovian dynamics between both simulations and discuss it with experimental data. We discuss the details in Secs. 4.2 and 4.3. Furthermore, we want to illustrate the biological relevance by applying it to a riboswitch design problem; this is shown in detail in Sec. 4.3.

4.2 Evaluation Setup

All benchmarks were performed on a machine with four *Intel(R) Xeon(R) CPU E5-2687W v2* processors with 3.4GHz running a Debian operating system. We implemented *StreAM* in Java; all sources are available in a GitHub repository[5]. The final implementation *StreAM-T_g* is integrated in a `Julia` repository [6]. We created plots using the `AssayToolbox` library for `R` [6,29]. We generate all random graphs using a generator for dynamic graphs[7].

Table 1: Details of the dynamic graphs obtained from MD simulation trajectories. $|V|$ is the number of vertices, $|E|$ the number of edges and δ_t is the average batch size of a simulation. We convert simulations to unit sphere dynamic graphs with $d \in [10,15]$Å.

	10Å	**11Å**	**12Å**	**13Å**	**14Å**	**15Å**	**Rand$_{g1}$**	**Rand$_{g2}$**	**Rand$_{g3}$**
$\|V\|$	65	65	65	65	65	65	500	500	500
$\|E\|$	94	129	189	241	298	353	500	1000	1200
δ_{avg}	6.1	15.6	19.4	18	19.6	23.8	80	100	120

Run-time Dependencies of StreAM on Adjacency Size. For every dynamic graph $G_t(V,E_t)$, we selected a total number of 100,000 snapshots to measure *StreAM* run-time performance. In order to perform benchmarks with increasing k, we chose randomly nodes $k \in [3,10]$ and repeated this 500 times for different numbers of snapshots (every 10,000 steps). We determined the slope (speed $\frac{frames}{ms}$) of compute time vs. k for random and MD graphs with different parameters (Tab. 1).

Run-time Dependence of StreAM on Batch Size. We measured run-time performance of *StreAM* for the computation of a set of all transitions $\mathscr{T}(V')$ with different adjacency sizes k as well as dynamic networks with increasing batch sizes. To test *StreAM* batch size dependencies, 35 random graphs were drawn with increasing batch size and constant numbers of vertex and edges. All graphs contained 100,000 snapshots and k is calculated from 500 random combinations of vertices.

Run-time Dependencies of StreAM-T_g on Network Size. We benchmarked the full computation of T_g with different $k \in [3,5]$ for increasing network sizes $|V|$. Therefore we performed a full computation with *StreAM*. *StreAM-T_g* sorts the obtained transition list, converts them into transition matrices and combines them into a global Markov model for each vertex.

[5] `https://github.com/BenjaminSchiller/Stream`

[6] `http://www.cbs.tu-darmstadt.de/streAM-Tg.tar.gz`

[7] `https://github.com/BenjaminSchiller/DNA.datasets`

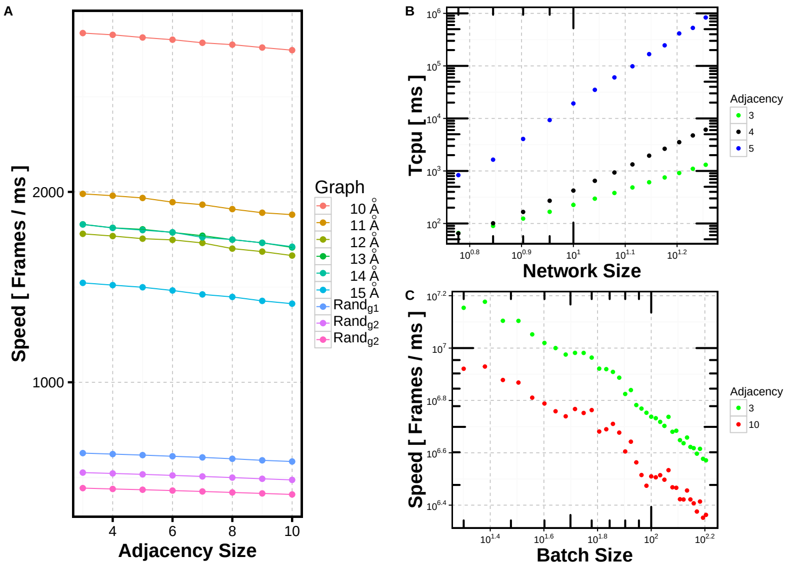


Fig. 3: Run-time performance of *StreAM*-T_g. **A**: Speed of computing a set of $\mathscr{T}(V')$ using *StreAM*.**B**: Performance of T_g full computation with increasing network size $|V|$ and different adjacency sizes $k = 3,4,5$. **C**: Speed of *StreAM* with increasing batch size for $k = 3,10$.

4.3 Run-time Evaluation

Figure 3B shows computational speeds for each dynamic graph. Speed decreases linearly with a small slope (Fig. 3A). While this is encouraging the computation of transition matrices for $k > 5$ is still prohibitively expensive due to the exponential increase of the matrix dimensions with $2^{\frac{k\cdot(k-1)}{2}}$. For G_t obtained from MD simulations, we observe fast speeds due to small batch sizes (Tab. 1).

Fig. 3B reveals that T_{cpu} increases linearly with increasing $|V|$ and with k exponentially. We restrict the T_g full computation to $k < 5$. In Fig. 3C, speed decreases linearly with δ_t. As δ_t represents the changes between snapshots our observation has implications for the choice of MD integration step lengths as well as trajectory granularity.

Application to Molecular Synthetic Biology. For both simulations of Sec. 2.3, we computed 17,039,360 transition matrices and combined them into 65 global models (one for each vertex of the riboswitch). To account for both the pair-interactions and potential stacking effects we focus on $k = 4$-vertex adjacencies and use dynamic RNA graphs with $d = 13$Å. One global transition matrix contains all the transitions a single nucleotide participates in. The stationary distribution and the implied entropy (changes) help to understand the effects of ligand binding and potential improvements on this (the design problem at hand). The ΔH obtained are shown in Fig. 4.

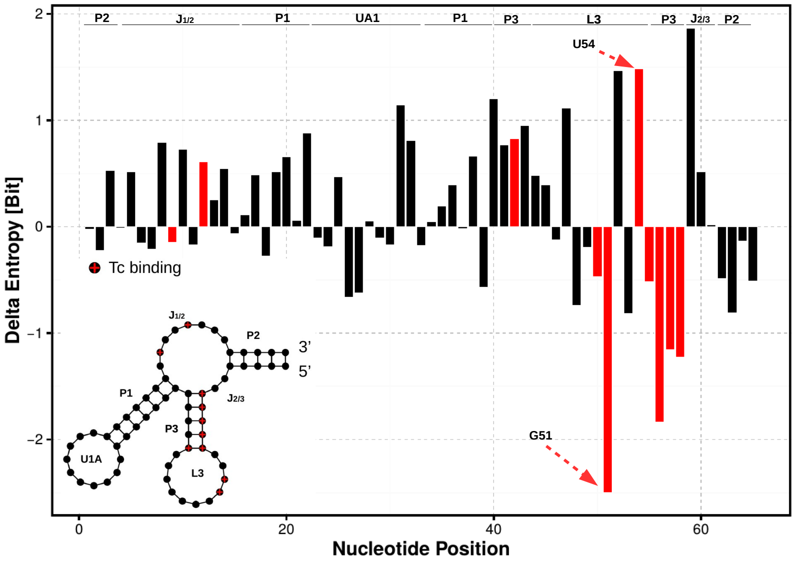


Fig. 4: ΔH for T_g of the native riboswitch and the one in complex with tetracycline (TC). Nucleotides with TC in complex are colored in red. At the top we annotate the nucleotides with secondary structure information.

A positive value of ΔH in Fig. 4 indicates a loss of conformational entropy upon ligand binding. Interestingly, the binding loop as well as complexing nucleotides gain entropy. This is due to the fact of rearrangements between the nucleotides in spatial proximity to the ligand because 70% of the accessible surface area of TC is buried within the binding pocket L3 [32]. Experiments confirmed that local rearrangement of the binding pocket are necessary to prevent a possible release of the ligand [21]. Furthermore crystallographic studies have revealed that the largest changes occur in L3 upon TC binding [32].

Furthermore, we observe the highest entropy difference for nucleotide G51. Experimental data reveals that G51 crosslinks to tetracycline when the complex is subjected to UV irradiation [4]. These findings suggest a strong interaction with TC and thus a dramatic, positive change in ΔH.

Nucleotides A52 and U54 show a positive entropy difference inside L3. Interestingly, molecular probing experiments show that G51, A52, and U54 of L3 are – in the absence of the antibiotic – the most modified nucleotides [32,13]. Clearly, they change their conformational flexibility upon ligand binding due they direct interaction with the solvent. U54 further interacts with A51,A52,A53 and A55 building the core of the riboswitch [32]. Taken together, these observations reveal that U54 is necessary for the stabilization of L3. A more flexible dynamics (ΔH) will change the configuration of the binding pocket and promotes TC release.

5 Summary, Conclusion, & Future Work

In this study, we demonstrate that *StreAM-T_g* fulfills our demands for a method to extract the coarse-grained Markovian dynamics of motifs of a complex RNA molecule. The effects observed in a designable riboswitch could be related to known experimental facts, such as conformational altering caused by ligand binding. Hence *StreAM-T_g* derived Markov models in an abstract space of motif creation and destruction. This allows for the efficient analysis of large MD trajectories. Thus we hope to elucidate molecular relaxation timescales, spectral analysis in relation to single-molecule studies, as well as transition path theory in the future. At present, we use it for the design of switchable synthetic RNA based circuits in living cells [8,7].

To broaden the application areas of *StreAM-T_g* we will extend it to proteins as well as evolutionary graphs mimicking the dynamics of molecular evolution in sequence space [17].

Acknowledgements. The authors gratefully acknowledge financial support by the LOEWE project compuGene of the Hessen State Ministry of Higher Education, Research and the Arts. Parts of this work have also been supported by the DFG, through the Cluster of Excellence cfaed as well as the CRC HAEC.

References

1. Alder, B.J., Wainwright, T.E.: Studies in Molecular Dynamics. J. Chem. Phys. 30, 459–466 (1959)
2. Aleksandrov, A., Simonson, T.: Molecular mechanics models for tetracycline analogs. J. Comp. Chem. 30(2), 243–255 (2009)
3. Andronescu, M., Condon, A., Hoos, H.H., Mathews, D.H., Murphy, K.P.: Computational approaches for RNA energy parameter estimation. RNA 16(12), 2304–2318 (2010)
4. Berens, C., Thain, A., Schroeder, R.: A tetracycline-binding rna aptamer. Bioorganic and Medicinal Chemistry 9(10), 2549 – 2556 (2001)
5. Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S., Karplus, M.: Charmm: A program for macromolecular energy, minimization, and dynamics calculations. J. Comp. Chem. 4(2), 187–217 (1983)
6. Buß, O., Jager, S., Dold, S.M., Zimmermann, S., Hamacher, K., Schmitz, K., Rudat, J.: Statistical Evaluation of HTS Assays for Enzymatic Hydrolysis of β-Keto Esters. PloS one 11(1) (2016)
7. Cameron, D.E., Bashor, C.J., Collins, J.J.: A brief history of synthetic biology. Nature reviews. Microbiology 12(5), 381–90 (2014)
8. Carothers, J.M., Goler, J.a., Juminaga, D., Keasling, J.D.: Model-driven engineering of RNA devices to quantitatively program gene expression. Science 334(6063), 1716–9 (2011)
9. Chodera, J.D., Noé, F.: Markov state models of biomolecular conformational dynamics. Curr. Opin. Struct. Biol. 25, 135 – 144 (2014)
10. Deigan, K.E., Li, T.W., Mathews, D.H., Weeks, K.M.: Accurate SHAPE-directed RNA structure determination. PNAS 106(1), 97–102 (2009)
11. Gan, H.H., Pasquali, S., Schlick, T.: Exploring the repertoire of RNA secondary motifs using graph theory; implications for RNA design. Nuc. Acids Res. 31(11), 2926–2943 (2003)

12. Hamacher, K., Trylska, J., McCammon, J.A.: Dependency map of proteins in the small ribosomal subunit. PLoS Comput Biol 2(2), 1–8 (2006)
13. Hanson, S., Bauer, G., Fink, B., Suess, B.: Molecular analysis of a synthetic tetracycline-binding riboswitch. RNA pp. 2549 – 2556 (2005)
14. III, T.E.C.: Simulation and modeling of nucleic acid structure, dynamics and interactions. Curr. Opin. Struct. Biol. 14(3), 360 – 367 (2004)
15. Jonikas, M.A., Radmer, R.J., Laederach, A., Das, R., Pearlman, S., Herschlag, D., Altman, R.B.: Coarse-grained modeling of large RNA molecules with knowledge-based potentials and structural filters. RNA 15(2), 189–99 (2009)
16. Laing, C., Schlick, T.: Computational approaches to RNA structure prediction, analysis, and design. Curr. Opin. Struct. Biol. 21(3), 306–318 (2011)
17. Lenz, O., Keul, F., Bremm, S., Hamacher, K., von Landesberger, T.: Visual analysis of patterns in multiple amino acid mutation graphs. In: Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on. pp. 93–102 (2014)
18. Manzourolajdad, A., Arnold, J.: Secondary structural entropy in RNA switch (Riboswitch) identification. BMC Bioinformatics 16(1), 133 (2015)
19. Parisien, M., Major, F.: The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. Nature 452(7183), 51–55 (2008)
20. Pronk, S., Páll, S., Schulz, R., Larsson, P., Bjelkmar, P., Apostolov, R., Shirts, M.R., Smith, J.C., Kasson, P.M., van der Spoel, D., Hess, B., Lindahl, E.: Gromacs 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. Bioinformatics 29(7), 845–854 (2013)
21. Reuss, A., Vogel, M., Weigand, J., Suess, B., Wachtveitl, J.: Tetracycline determines the conformation of its aptamer at physiological magnesium concentrations. Biophysical Journal 107(12), 2962 – 2971 (2014)
22. Schiller, B., Jager, S., Hamacher, K., Strufe, T.: Stream - a stream-based algorithm for counting motifs in dynamic graphs. In: Dediu, A.H., Quiroz, F.H., Martín-Vide, C., Rosenblueth, D.A. (eds.) Algorithms for Computational Biology, AlCoB 2015, Mexico City, Mexico, August 4-5, 2015, Proceedings. LNCS, vol. 9199, pp. 53–67. Springer (2015)
23. Schlick, T.: Mathematical and Biological Scientists Assess the State of the Art in Rna Science At an Ima Workshop, Rna in Biology, Bioengineering, and Biotechnology. International Journal for Multiscale Computational Engineering 8(4), 369–378 (2010)
24. Schrödinger, L.: The PyMOL molecular graphics system, version 1.8 (2015)
25. Senne, M., Trendelkamp-schroer, B., Noe, F.: EMMA: A Software Package for Markov Model Building and Analysis. J. Chem. Theory Comput. (2012)
26. Shapiro, B.A., Yingling, Y.G., Kasprzak, W., Bindewald, E.: Bridging the gap in RNA structure prediction. Curr. Opin. Struct. Biol. 17(2), 157–165 (2007)
27. Spedicato, G.A.: markovchain: discrete time Markov chains made easy (2015), R package version 0.4.3
28. Stombaugh, J., Zirbel, C.L., Westhof, E., Leontis, N.B.: Frequency and isostericity of RNA base pairs. Nucleic Acids Research 37(7), 2294–2312 (2009)
29. Team, R.D.C.: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008)
30. Tung, C.S.: RNA Structural Motifs. Life Sciences pp. 1–4 (2002)
31. Wunnicke, D., Strohbach, D., Weigand, J.E., Appel, B., Feresin, E., Suess, B., Muller, S., Steinhoff, H.J.: Ligand-induced conformational capture of a synthetic tetracycline riboswitch revealed by pulse EPR. RNA 17(1), 182–188 (2011)
32. Xiao, H., Edwards, T.E., Ferré-D'Amaré, A.R.: Structural basis for specific, high-affinity tetracycline binding by an in vitro evolved aptamer and artificial riboswitch. Chemistry & Biology 15(10), 1125 – 1137 (2008)