i-ViDa: Visualizing Energy Landscapes and Trajectories of DNA Reactions

Chenwei Zhang and Yibo Jiao

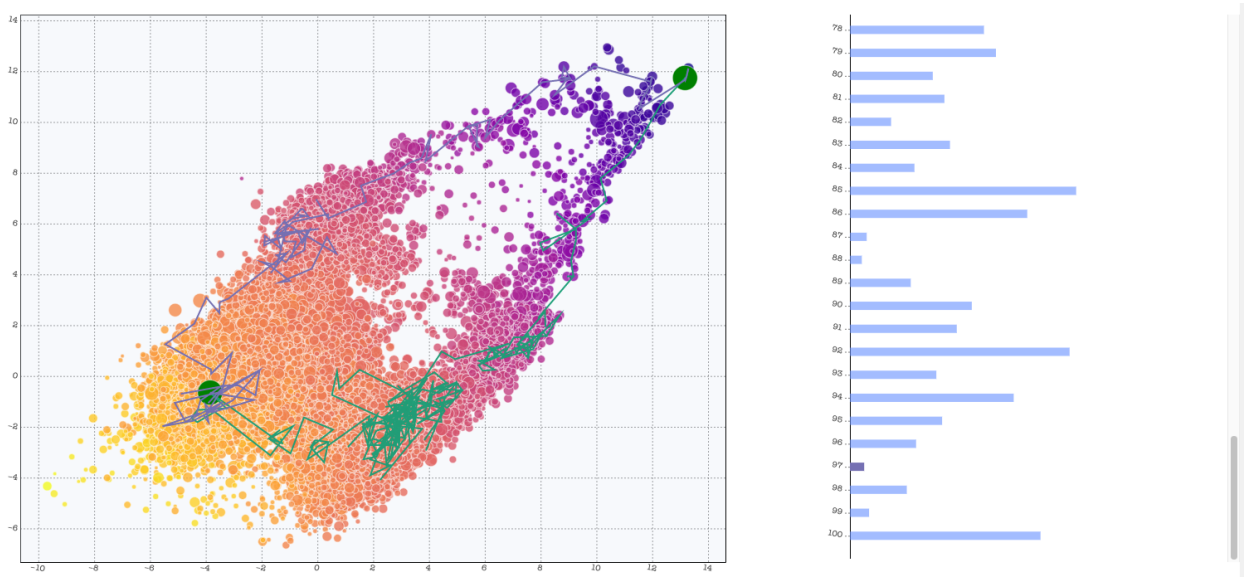

Fig. 1. Our current implementation.

1 Introduction

DNA nanotechnologies have been extensively studied owing to potential applications such as DNA beacons [7] and artificial neural networks [5]. DNA reaction mechanisms, however, are still not well understood. There is an immediate need to have reliable solutions to help synthetic biologists and molecular programmers better understand the mechanism of reactions of interest, so that further to help design novel nucleic acid reactions with more promising applications.

It turns out that visualizing reaction energy landscapes and trajectories provides a meaningful way to study reaction mechanisms. An energy landscape is comprised with a set of secondary structures visited in the sampled trajectories. A reaction trajectory is depicted as a sequence of secondary structures from the reactants to the products of a DNA reaction, along with the reaction time to transition from one secondary structure to the next. A secondary structure describes a set of strands with their base pairs (bp) formed via hydrogen bonding in terms of Watson-Crick and/or Wobble base pair rules, and each secondary structure has an associated free energy that is determined by underlying thermodynamic parameters. The DNA reaction process is stochastic so that reaction trajectories could be modelled based on the continuous-time Markov chain, by which each transition between states (secondary structures) in a trajectory shows an elementary step with a single base pair forming or breaking.

There are some previously-published visualization methods for energy landscapes and reaction trajectories. Schaeffer et al. [6] designed the Multistrand simulator to output secondary structures with the dot-parenthesis (dp) notation, and a sequence of such secondary structures from the initial to finial structures represent a trajectory. However, this way is constrained by only showing one single trajectory and does not allow situating the trajectory on the energy landscape. Machineck et al. [4] used a coarse-grained method to show energy landscapes and lay out the reaction trajectories on the landscapes, while the interpretation of the coarse-grained plots is difficult due to the lack of explicit state information. Flamm et al. [3] used barrier trees to visualize energy landscapes and Castro et al. [2] proposed a deep graph embedding model to map secondary structures into low-dimensional space to uncover the energy landscape. However, both of the two approaches do not address showcasing reaction trajectories on such landscapes. Accordingly, having a well-designed visualization tool for energy landscapes and trajectory plots is imminent.

This project is an extension of my RPE project, for which I designed an visualization tool based on a deep graph embedding approach to map high-dimensional DNA secondary structures into low-dimensional space to show energy landscapes, and then lay out different trajectories on the landscape. Although there was a tooltip design for displaying secondary structures and their corresponding information, this tool is limited for explicit comparison of different states and trajectories, and it does not integrate reaction time into the plots. Moreover, my RPE has not addressed a suitable way to quantitatively evaluate the graph embedding.

To tackle these limitations, the purpose of the course project is to design a user-friendly interactive visualization tool, that we named **i-ViDa**, in the shape of a website, to allow users to manipulate the visualization of energy landscapes and trajectories of interest. We expected that by using our designed tool, users can easily address these questions:

- Which trajectories are the most important ones for the reaction? In other word, which trajectories are dominant in the reaction?
- How many significant reaction pathways are existing from the

- *Chenwei Zhang. E-mail: cwzhang@cs.ubc.ca*
- *Yibo Jiao. E-mail: jyibo@cs.ubc.ca*

initial to final states for the reaction?

- For a specific trajectory, how is the total reaction time related to the transition steps?
- For a specific trajectory, how do the state energy change over the transition steps and times?
- Can users identify the traps or barriers for the reaction, and what are the states' information for these traps?
- What are important states read from the visualization? Specifically, what states have the most reaction trajectories passed by? Additionally, what are some likely trajectories that start from a certain state?

Answering the above questions can help evaluate the visualization tool. We will concretely describe these questions in Section 3.2.

Additionally, in this project we also plan to design a "distance" metric to precisely evaluate the graph embedding approach. Although using the interactive visualization tool we can qualitatively assess the embedding approach and implement a set of analyses based on the input embedding datasets, we would like to have a more accurate way to quantify the embedding model. Specifically, we would like to find a reasonable metric that can measure the preservation of local and global structure, thereby inferring the performance of the embedding.

2 Related Work

2.1 ViDa

This course project builds on a visualization model, ViDa, proposed in my RPE project. ViDa is a new approach to visualize energy landscapes and trajectory plots in light of a deep graph embedding approach. The framework of ViDa is shown in Figure 2, which consists of five major parts: the Multistrand simulator that is to produce secondary structure and trajectory information, the convertor that converts secondary structures represented by dp notations to adjacency matrices, a deep embedding model called GSAE, an additional dimensionality reduction technique such as PCA and PHATE, and an interactive plotting tool that has a tooltip feature using Plotly in Python. The input of ViDa is a reaction with a sequence and initial and final secondary structures, and the output is a visualization plot. ViDa has been demonstrated to embed high-dimensional secondary structures into low-dimensional Euclidean space to display energy landscapes and to lay out meaningful trajectories in the landscapes. Using ViDa users can easily retrieve the state information such as secondary structures, energies, reaction times, hairpin information, and so on, from the energy landscape. However, due to the limitation of the interactive plotting tool, it is not very straightforward to compare different trajectories of a reaction and find potential states that may affect the reaction process. Additionally, kinetic traps' information is restricted by ViDa. Although ViDa shows time information in the tooltip, it does not allow time as a variable while plotting. Therefore, in the course project we plan to improve ViDa to enable a more comprehensive and accurate analysis of simulated reaction trajectories and energy landscapes, further to well understand reaction mechanisms.

3 Data and Task Abstraction

In this section, we introduce the data abstraction and task abstraction. The datasets we used for this project were generated from my RPE project and converted to particular formats for further encode.

3.1 Data Abstraction

Currently, there are two reactions of interest, each of which has two tables, as shown in Table 1 and 2. (In the future we may have more datasets for different types of reactions of interest, but the data type and dataset type are the same as the following.)

We denote our dataset as:

$$\begin{aligned} S &= \{S_1, ..., S_{46606}\} \\ T &= \{T_1, ..., T_{100}\} \end{aligned} \tag{1}$$

For an arbitrary trajectory T_i of length m, i.e., T_i has m reaction steps:

$$\begin{aligned} T_i &= \{I_i, R_i\} \\ I_i &= [I_{i1}, ..., I_{ij}, ..., I_{im}] \\ R_i &= [R_{i1}, ..., R_{ij}, ..., R_{im}] \end{aligned} \tag{2}$$

where I_i, R_i are ordered lists of indices and time of the states in trajectory T_i, respectively.

In one specific reaction, the first table has 46606 items and ten attributes including the secondary structure represented by the dot-parenthesis notation, the coordinate, the energy, the average reaction time (simulation time, not the real wall-clock time) of each item, the occupancy density of each secondary structure, i.e. how many different trajectories pass through this structure over the total trajectories. For a state S_j, the occupancy density of S_j, $d(S_j)$, is expressed as:

$$\begin{aligned} d(S_j) &= \sum_{i=0}^{100} b(S_j, i) \\ b(S_j, i) &= \begin{cases} 1, & if\ ID(S_j) \in I_i \\ 0, & otherwise, \end{cases} \end{aligned} \tag{3}$$

and five secondary structure-related information. Specifically, these five attributes include the size of hairpin structures formed in each of the strands, i.e. the number of intra-strand base pairs in each complementary strands, the number of correctly bound inter-strand base pairs, the total number of inter-strand base pairs, and a label indicating whether the complementary strands possess at least one inter-strand base pair, namely, label "0" refers to two strands separated (disconnected) and label "1" refers to two strands bound (connected).

The other table has 100 items and four attributes. Each item refers to a trajectory. One attribute is the index which is made up of a list. The element of the list refers to the index of each item in the first table. Using the indices from the list to retrieve the items, we can get a series of secondary structures and their corresponding dot-parenthesis notations, coordinates, and energies in a trajectory for further visualization. The second attribute is the transition time, which is different from the average reaction time in the previous table. Because the DNA reaction is stochastic, every simulated trajectory consists of different states and the transition time from one state to the next is not deterministic. The average reaction time of each state is calculated by averaging all transition times of that state in the reaction. The third and forth attributes are the reduced trajectory indices and cumulative transition times of the states that are remained after filtering. There are some trajectories are extremely long. How can we find a meaningful way to reduce the length of the trajectory but still preserves its essential information? To solve this problem, we conceive a high-density pass filter to filter out the states with low occupancy densities in a trajectory, while the rest of states with high occupancy densities are preserved and the cumulative reaction time for each of them were recorded. By means of this filter, we can significantly reduce the number of states in a trajectory, for the purpose of the arc diagram design.

Table 1. Data abstraction for secondary structure information.

Attribute	Type	Range
ID	categorical	[1, 46606]
DP notation	categorical	N/A
Coordinate X	quantitative	[-9.7, 13.3]
Coordinate Y	quantitative	[-6.7, 12.9]
Energy	quantitative	[-39.47, 10.87]
Average time	quantitative	[0, 3.60 e-8]
Occupancy density	ordinal	[1, 100]
Intra-strand bp (left)	quantitative	[0, 12]
Intra-strand bp (right)	quantitative	[0, 12]
Corrected inter-strand bp	quantitative	[0, 25]
Total inter-strand bp	quantitative	[0, 25]
Binding	categorical	{0, 1}

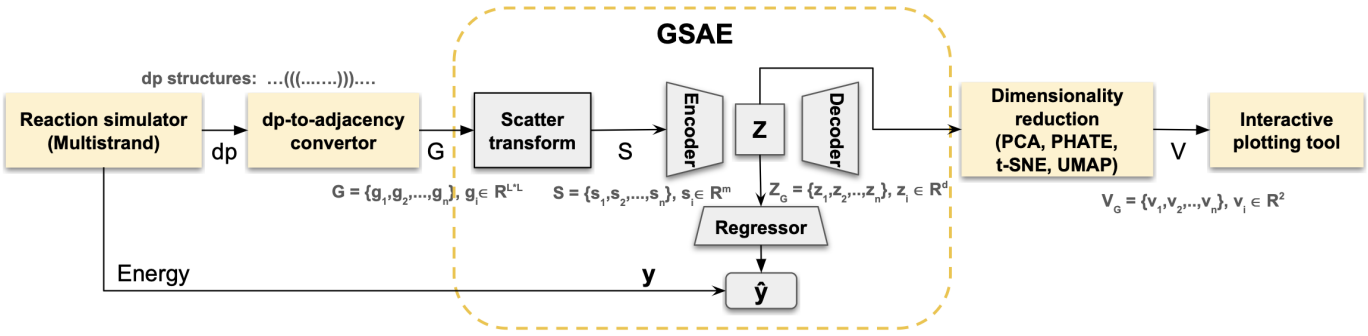

Fig. 2. The framework of ViDa. The scattering transform and semi-VAE make up the GSAE model. $\hat{y}$ is the predicted energy by the regressor network and y is the real energy produced by the simulator.

Table 2. Data abstraction for trajectory information.

Trajectory	Type	Cardinality
Index	quantitative	[104, 54762]
Time	quantitative	[104, 54762]
Reduced index	quantitative	[TBD, $\sim$ 100]
Cumulative time	quantitative	[TBD, $\sim$ 100]

3.2 Task Abstraction

3.2.1 Who

The fellow tool builders will be me and my partner. The gatekeeper will be my supervisor, Anne, and the front-line analysts will be the domain expert, Erik, who works on DNA computing students, who yet have not been determined.

3.2.2 Action and Target

We present our task abstractions by introducing three levels (analyze, search, and query) of actions and targets in separated spaces which includes the trajectory space and the state space. Each space represents one of the two datasets discussed in the data abstraction section, respectively.

Trajectory-space-related Tasks In *analyze* level, we expect the users to be able to:

(i) present any trajectory on 2D.

(ii) discover correlations and features of trajectories. In lower-level, the users can answer questions such as:

- Do the total time and steps of trajectories correlate?
- Do the total time/steps and spatial shape on reduced dimension correlate?

(iii) present the energy flow of trajectories.

(iv) present the occupancy density flow of trajectories.

(v) derive the approximate number of reaction traps in trajectories visually.

In *search* level, the users can:

(vi) look up/explore trajectories with various features. Features of interest might include long/short of reaction time/steps.

In *query* level, the users can:

(vii) identify reaction traps of trajectories via a visual, topological or statistical method.

(viii) compare features of trajectories. In lower-lever, the following questions might be interesting to users.

- Do trajectories with similar time/steps have similar energy/occupancy density flow?
- Do rare/common reaction trajectories differs in reaction time/steps? We define the rareness of trajectory T_i of length m, $r(T_i)$ as:

$$r(T_i) = \frac{\sum_{j=0}^{m} density(S_{I_{ij}})}{m} \tag{4}$$

  i.e., the more states of higher occupancy density that T_i contains, the more common T_i is.
- Do trajectories that follow similar spatial shapes in 2D Euclidean space have similar energy flows/typologies?

(ix) summarize all trajectories' energy flow, occupancy density flow, or occupancy density distribution.

State-space-related tasks In *analyze* level, our visualization helps the users to:

(x) present states on 2D.

(xi) discover the state distribution over energy, time, and some special structures.

(xii) present secondary structure of states.

(xiii) drive state coverage rates of trajectories with states of given density. We define the coverage rate of a trajectory with given density is the number of states with higher/equal given density divided by the total number of states in the trajectory. For trajectory $T_i = \{I_i, R_i\}$ of m steps in total, i.e., $|I_i| = m$, the coverage rate of T_i with a density threshold k, $d(T_i,k)$ is:

$$d(T_i,k) = \frac{\sum_{j=0}^{m} f(S_{I_{ij}},k)}{m}$$
$$f(S_{I_{ij}},k) = \begin{cases} 1, & if\ density(S_{I_{ij}}) \geq k \\ 0, & otherwise. \end{cases} \tag{5}$$

In lower-level, the users might question: what is a reasonable threshold for filtering states with higher occupancy densities to cover the most of steps of a long trajectory? This is also a task relates to trajectory-space.

In *search* level, the users can:

(xiv) look up/explore states with various features. Features of interest might include: high/low energy, reaction time and occupancy density; type/number of links in the secondary structure.

In *query* level, the users can:

(xv) compare features of states. In lower-level, our tool answers the following questions:

- Do states with similar reaction time/energy have similar structures?
- Do states with higher densities have special structures compared to rare states?
- Do states with certain structures cause the kinetic traps?

4 Solution

4.1 Implementation

We use D3.js as our framework for this project [1]. There is no pre-existing software were dependent.

4.2 Scenario

We propose our potential solutions by introducing scenarios of use aligned with each corresponding task discussed in Section 3.2.

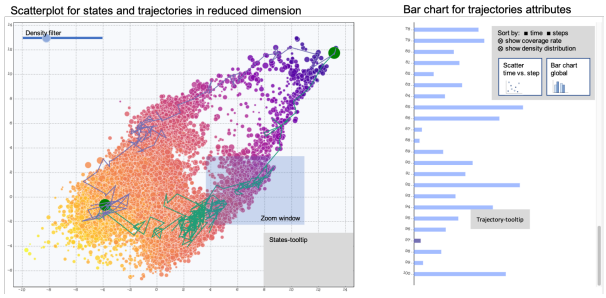

Fig. 3. Potential solution for overview-view design

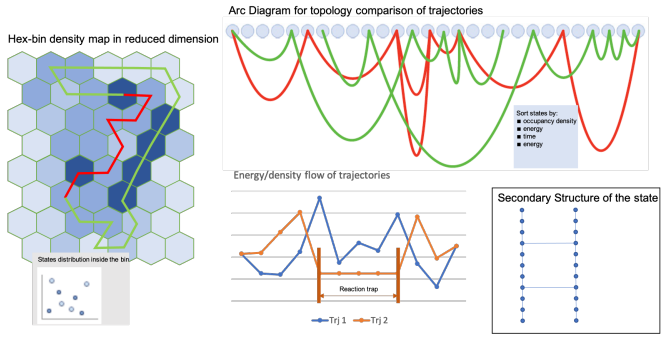

Fig. 4. Potential solution for compare-view and utility-view design

For each scenario aligned with each tasks discussed above, to perform the tasks, the users can:

(i) click any line mark on the large bar chart to encode a trajectory, in which the spatial shape in 2D on the large scatter plot is presented for the selected trajectory.

(ii) select multiple trajectories with different/similar time/steps on the bar chart and view their spatial shapes on the large scatter plot. The users can view correlation between time and steps in the small scatter plot. The users can also view overall ranking of the selections in the small bar chart, which is a redundant visualization of the large bar chart but contains all trajectories in one window.

(iii) click any line mark on the bar chart and view the energy flow on the line chart.

(iv) click any line mark on the bar chart and view the density flow on the line chart.

(v) click any line mark on the bar chart to select a typical trajectory, and visually count the number of clusters of dense line segments in this selection.

(vi) display the bar chart with a selected attribute (time or steps) by clicking the button on the top right of the large bar chart. The users can also sort the trajectories based on the selected time or steps.

(vii) visually locate dense clusters of line segments in the large scatter plot. Topologically, the users can look up dense arcs segments in the arc diagram Figure 4. Statistically, the users can identify whether a trap exists in terms of whether a flat portion in a energy flow curve exists, as shown in Figure 4.

(viii) click line marks on the bar chart to select trajectories for comparison. The users can judge the differences of marks in the energy flow. Recall that trajectories with more states of higher occupancy densities are common trajectories and vice versa, the users can show density distribution of trajectories by clicking the button on the large bar chart. The hexbin chart encodes all data in a coarse level, and it is also a redundant visualization of the large scatter plot. The users can use the hexbin chart to view the spatial shapes of trajectories without dense line segment crossings.

(ix) click the option button on the large bar chart to enable line marks to show attributes required in the task. Another solution is to construct a matrix of colormap for all trajectories.

(x) view the large scatter plot.

(xi) use the zoomed window in the large scatter plot. Since our data is generated in a way that states with similar structures are closed together in 2D, states that fall inside a smaller selected window intend to have more similar structures. The users can also click on any bin inside a hexbin chart to view distribution because states falls inside a bin also have similar structures with similar reasoning.

(xii) click any point mark in the large scatter plot and view its actual secondary structure in the visualization component on the right bottom corner in Figure 4.

(xiii) use the slider on the large scatter plot to manipulate the filtering of states. Trajectories with filtered states will have less information compared to the original ones.

(xiv) use the zooming function in the large scatter plot to view interested features.

(xv) perform this task similarly as task (xi) and combine with task (xii).

5 Milestones

- Preliminary work, Sep.21 - Sep.30
  - (3 hours, both) introduce project background knowledge introduction, dataset discussion, visualization methods discussion
  - (3 hours, both) project pitch preparation
  - (5 hours, Chenwei) generate and translate datasets into certain formats
- Project proposal, Oct.1 - Oct.21
  - (6 hours, both) discuss all possible project tasks
  - (5 hours, both) discuss potential designed prototypes
  - (3 hours, Chenwei) connect with domain experts to discuss the proposal
- Implementation before update report, Oct.24 - Nov.15

  - (3 hours, Chenwei) provide ideas and layout for the interactive tool design
  - (5 hours, Yibo) complete interaction functionality of overview-view
  - (3 hours, Yibo) complete basic functionality of hexbin chart
  - (6 hours, Yibo) complete basic functionality of arc diagram
  - (3 hours, Yibo) complete basic functionality of flow chart
  - (2 hours, both) finalize interaction activities to implement
  - (10 hours, Yibo) complete interaction functionality of flow chart
  - (5 hours, Chenwei) give feedback to Yibo's design and discussion
  - (10 hours, Chenwei) read some evaluation papers for embedding approaches and get insights from them
- Implementation deadline, Nov.15 - Dec.7
  - (3 hours, both) discuss feedback from peer-review, finalize required functionality
  - (10 hours, Yibo) complete all functionality of all views.
  - (5 hours, both) discuss styling choices.
  - (10 hours, Yibo) styling, debugging, deploying the final version of implementation
  - (10 hours, Chenwei) Design the meaningful evaluation metric to quantify the embedding data
- Finalization, Dec.7 - Dec.16
  - (10 hours, both) prepare the final presentation
  - (20 hours, both) document all results and write the final report

6 Discussion

7 Future Work

8 Conclusion

References

[1] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

[2] E. Castro, A. Benz, A. Tong, G. Wolf, and S. Krishnaswamy. Uncovering the folding landscape of RNA secondary structure using deep graph embeddings. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4519–4528. IEEE, 2020.

[3] C. Flamm, I. L. Hofacker, P. F. Stadler, and M. T. Wolfinger. Barrier trees of degenerate landscapes. *Zeitschrift für Physikalische Chemie*, 216(2):155–155, 2002.

[4] R. Machinek, T. Ouldridge, N. Haley, J. Bath, and A. Turberfield. Programmable energy landscapes for kinetic control of DNA strand displacement. *Nature Communications*, 5, 2014.

[5] L. Qian and E. Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196–1201, 2011.

[6] J. M. Schaeffer, C. Thachuk, and E. Winfree. Stochastic simulation of the kinetics of multiple interacting nucleic acid strands. In *International Workshop on DNA-Based Computers*, pages 194–211. Springer, 2015.

[7] K. Wang, Z. Tang, C. J. Yang, Y. Kim, X. Fang, W. Li, Y. Wu, C. D. Medley, Z. Cao, J. Li, P. Colon, H. Lin, and W. Tan. Molecular engineering of DNA: molecular beacons. *Angew Chem Int Ed Engl*, 48(5):856–870, 2009.