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

Chenwei Zhang and Yibo Jiao

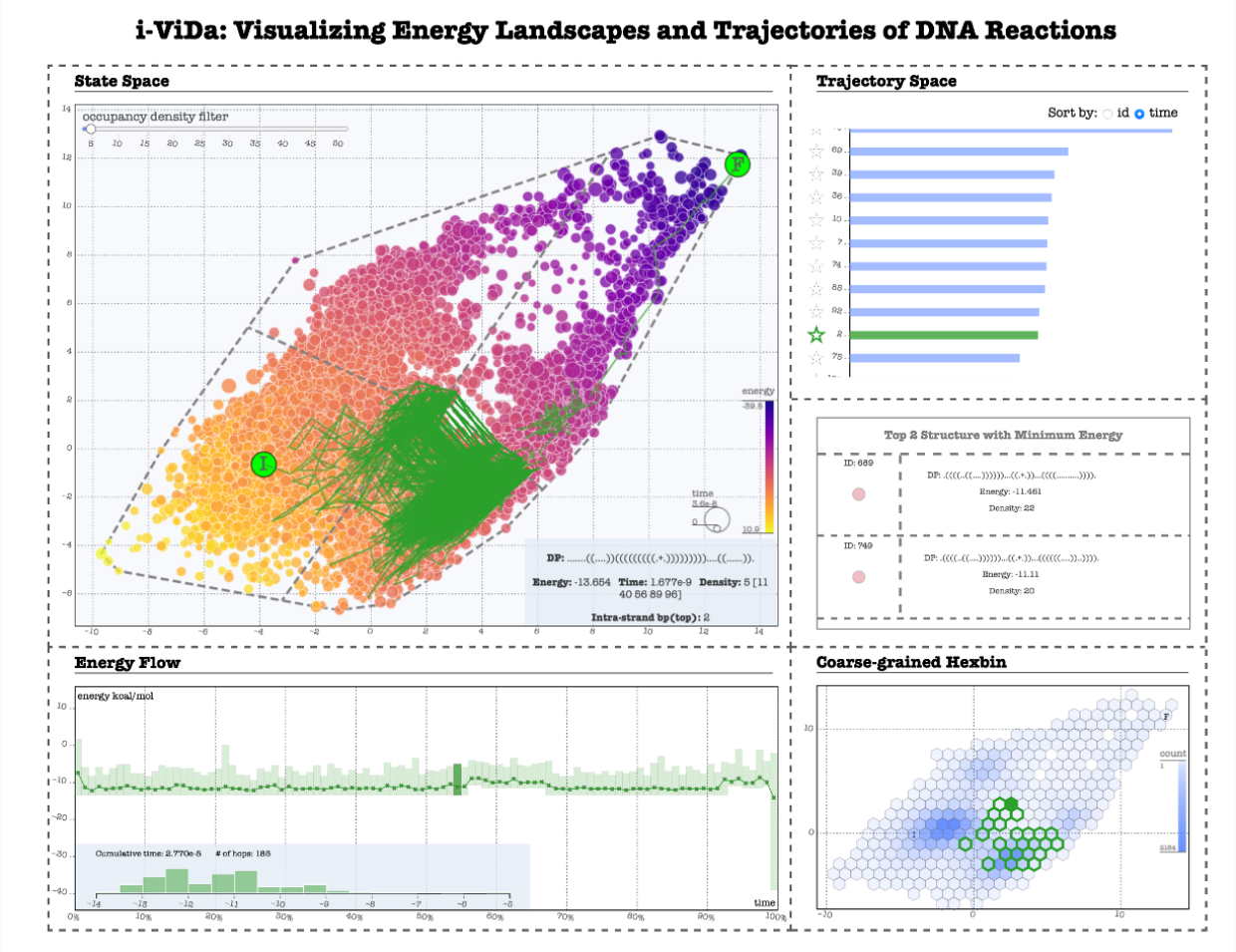


Fig. 1. The thumbnail of i-ViDa.

1 Introduction

In the past few decades, DNA and RNA nanotechnologies have been developed that are capable of sensing and responding to changes in their environments, self-assembling into complex structures, and simulating computational models. The behaviour of these technologies depends on nucleic acid thermodynamics (which can be used to predict properties of nucleic acid systems in equilibrium) and kinetics (which predicts rates of change and folding dynamics). Thermodynamics of nucleic acids has been extensively studied, but the mechanisms that influence nucleic reaction kinetics are less 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.

In this project, we are visualizing DNA hybridization reactions, in which one unpaired DNA strand hybridizes with its complementary strand to form a fully paired duplex. It is an extension of Chenwei's RPE project, for which he designed a visualization tool, **ViDa**, 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 to the explicit comparison of different states and trajectories, and it does not integrate reaction time into the plots.

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. i-ViDa is an additional layer built on top of ViDa, replacing the previously Plotly-made interactive plotting tool, which allows users to plot latent space produced by ViDa, and

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

then manipulate the visualization of energy landscapes and trajectories of interest.

We expected that by using our designed tool, users can easily address these six questions:

Q1: Which trajectories are the most important ones for the reaction? In other words, which trajectories are dominant in the reaction?

Q2: How many significant reaction pathways are existing from the initial to final states for the reaction?

Q3: For a specific trajectory, how does the state energy change over the transition times? In the following discussion, we call this energy change over the transition times as the energy flow.

Q4: For a specific trajectory, how does the occupancy density change over the transition times? In the following discussion, we call this occupancy density change over the transition times as the occupancy density flow.

Q5: Can users identify the traps or barriers for the reaction, and what is the states' information for these traps?

Q6: 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 5.2.

2 Background

To analyze reaction mechanisms, we are interested in visualizing secondary structures in the energy landscape and the trajectory space.

Secondary structure 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. Conventionally, secondary structures are represented by dot-parenthesis notations.

Dot-parenthesis notation Dot-parenthesis (dp) notation is a simple way to represent a secondary structure of DNA or RNA. Each character represents a base. Dots indicate unpaired bases and matching parentheses indicate paired bases. The number of open and closed parentheses is always equal. The symbol "+" in the dp notation separates strands. For example, in the dp notation $3'$-...(((...-$5'$+$3'$-...)))...-$5'$ for the secondary structure of two DNA strands A ($3'$-TGACGATCA-$5'$) and $\bar{A}$ ($3'$-TGATCGTCA-$5'$), the left part of the "+" sign corresponds to strand A and the right part corresponds to strand $\bar{A}$. Three open parentheses indicate that the bases "CGA" in strand A are paired with the bases "TCG" in strand $\bar{A}$ which are represented by three closed parentheses.

Reaction trajectory A reaction trajectory is depicted as a sequence of secondary structures from the initial to the final of a DNA reaction, along with the reaction time to transition from one secondary structure to the next.

Reaction mechanism The DNA reaction process is stochastic so that reaction trajectories could be modeled 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.

Energy landscape An energy landscape is comprised of a set of secondary structures visited in the sampled trajectories.

State and trajectory space In this project, we are aiming to visualize the secondary structures on the latent space, so we name each of them as states and the latent space as state space. Then we lay out the reaction trajectories on the state space, which we called as trajectory space.

Kinetic traps In a specific DNA hybridization reaction, if there exist some intermediate reaction states with certain secondary structures, such as stable hairpins, the reaction rate will be significantly reduced, then we call these intermediate reaction states as kinetic traps (or barriers). The trap's energy is usually the local minimum.

3 Related Work

There are some previously-published visualization methods for energy landscapes and reaction trajectories. Schaeffer et al. [7] 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 final 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. [5] 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 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 necessary.

Describing and visualizing high-dimensional data usually relies on some dimensionality-reduction approaches to reduce significantly the number of features in the inputted data, while retaining the major information. Some common approaches can be categorized into two classes: linear such as PCA [4] and non-linear such as PHATE [6] methods. PCA favors learning global structure of high-dimensional data while PHATE is designed for keeping both global and local structures and it achieves great success in many kinds of datasets. Principle component analysis (PCA) is a matrix factorization-based linear dimensionality reduction approach. The basic idea of PCA is to project the initial input data features onto a small number of new features that are linear combinations of the initial features. PHATE, standing for potential of heat diffusion for affinity-based transition embedding, is a newcomer for embedding high-dimensional data onto a lower-dimension space, normally onto 2-D Euclidean space. PHATE is a nonlinear and unsupervised method designed to capture both local and global structure between inputted high-dimensional datapoints.

4 Previous System: ViDa

This course project builds on a visualization model for DNA reaction trajectories, ViDa, proposed in Chenwei's RPE project. ViDa is a new approach to visualizing 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 add an additional layer on top of ViDa to replace the current interactive plotting tool, therefore improving ViDa to enable a more comprehensive and accurate analysis of simulated

reaction trajectories and energy landscapes, further to well understand reaction mechanisms.

5 Data and Task Abstraction

In this section, we introduce data abstraction and task abstraction. The datasets we used for this project were generated from the ViDa model and then restructured and converted to particular formats for further encoding.

5.1 Data Abstraction

Currently, there are two reactions of interest, each of which has two tables, as shown in Table 1 and 2. The first table describes the state space, in which each node represents a state with a dp-notation secondary structure accompanied by some related information. The second table describes the trajectory space, in which each trajectory is a series of hops through the state space. We denote our dataset as:

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

where S refers to a set of states and T refers to a set of trajectories. 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 times of the states in the trajectory T_i, respectively.

In our chosen reaction, the first table has 46606 items and eleven attributes including the secondary structure represented by the dot-parenthesis notation, the coordinate (2D X/Y coordinate in the dimensionally reduced space), the energy, the average reaction time (simulation time, not the real wall-clock time) of each item that is calculated by averaging all transition times of that state in the reaction, 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 six secondary structure-related information. Specifically, these six attributes include the number of intra-strand base pairs in each complementary strand, 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), and the class label that each state belongs to.

Since the state space is relatively large, how can we find a meaningful way to reduce the state space size but still preserve its essential information? To solve this problem, we conceive a high-density pass filter to filter out the states with low occupancy densities (see Eq. 3), while the rest of the states with high occupancy densities are preserved. By means of this filter, we can significantly reduce the number of nodes in the state space for the purpose of controlling the overview of the main scatter plot, and the state size can be reduced from 54762 to a minimum of 104.

The other table has 100 items and two attributes. Each item refers to a trajectory with time information. 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 shortest and longest times of the trajectories are 1.5e-7 s and 1.5e-4 s, respectively. Additionally, we are also interested in judging whether a trajectory is common or rare, thereby, we define the rareness of trajectory T_i of length m, $r(T_i)$ as:

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

i.e., the more states of relatively high occupancy density that T_i contains, the more common T_i is.

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 (top)	quantitative	[0, 7]
Intra-strand bp (bot)	quantitative	[0, 7]
Corrected inter-strand bp	quantitative	[0, 25]
Total inter-strand bp	quantitative	[0, 25]
Binding	categorical	{0, 1}

Table 2. Data abstraction for trajectory information.

Trajectory	Type	Range
List of indices	quantitative	
(List length of indices, m)		[104, 54762]
List of times	quantitative	
(List length of times, m)		[104, 54762]

5.2 Task Abstraction

i-ViDa aims to support users in visualizing the embedding data and then getting insight from the visualization by answering the questions addressed in Section 1.

5.2.1 Who

The gatekeeper will be my supervisor, Dr. Anne Condon, and the front-line analysts will be the domain expert, Dr. Erik Winfree, who works on DNA computing studies.

5.2.2 State-space-related Tasks

In state space, we expect i-ViDa can help users to be able to accomplish these tasks:

T-S1 See the latent space of secondary structures and retrieve the selected state with various features. Features of interest include the secondary structure with dp-notation, energy, average reaction time, occupancy density, and structure-related information.

T-S2 See the reduced latent space by manipulating some controllable parameter.

T-S3 Aggregate similar states into one group to see coarse-grained latent space.

Fig. 2. The framework of ViDa. It includes five major parts: the Multistrand simulator which is used to generate stochastic trajectories of the given reaction; a convertor to convert a dp-notation secondary structure to a graph represented by an adjacency matrix; a graph deep embedding model called GSAE which is embedding the input graphs to a relatively low dimension latent space; further dimensionality reduction algorithms such as PCA and PHATE, which is used to project the GSAE-embedding features to 2D Euclidean space; a Plotly-made interactive plotting tool.

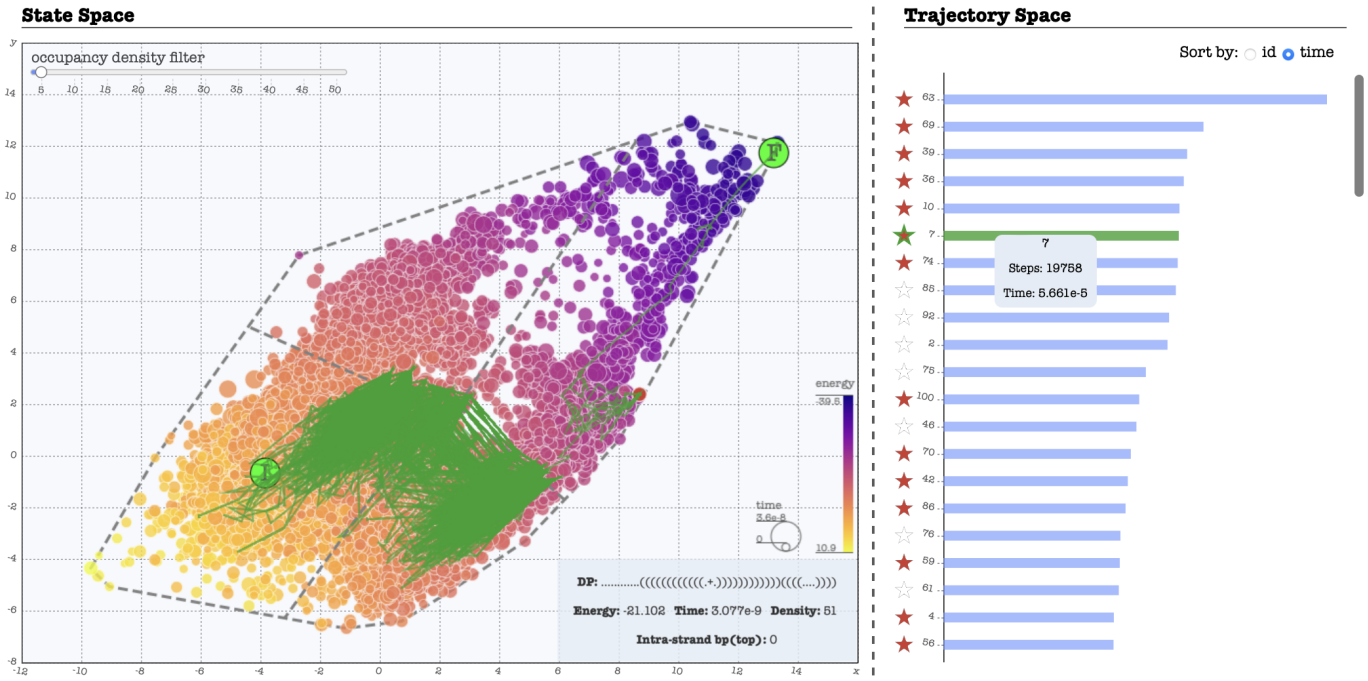


Fig. 3. The state space and trajectory space panes.

5.2.3 Trajectory-space-related Tasks

In trajectory space, we expect i-ViDa can help users to be able to accomplish these tasks:

T-T1 See simulated trajectories in latent space and retrieve the related information such as the reaction time of specific trajectories.

T-T2 Compare the spatial shapes of trajectories with notably different reaction times.

T-T3 Compare different trajectories in terms of their corresponding energy flows.

T-T4 Identify the number of kinetic traps in trajectories and capture the traps' information including their secondary structures, energies, average reaction times, and so on.

T-T5 Identify the significant reaction pathways, i.e. justify what types of trajectories are dominant in the reaction, and summarize the major reaction pathways based on the trajectories.

6 Solution

We use D3.js as our framework for this project [1], which is distinct from ViDa's interaction tool that was made using the Plotly library in Python.

6.1 Idiom Choices

In this section, we introduce four major designed views in i-ViDa and how the interaction works through these four views.

State space and trajectory space panes Fig. 3 shows the state space and trajectory space panes. The state space pane is a scatter plot, in which each circle mark encodes a state. Color and radius channels of the circle marks encode states' energy and average time, respectively. The initial and final states are encoded by the green circles marked I and F. Each colored line mark represents a selected trajectory. A density slider that is placed on the top left corner allows users to slide to an arbitrary occupancy density threshold, reducing the size of the state space by filtering out those states that occupy less than the threshold density. A tooltip box is placed on the bottom right corner to display the selected state with dp-notation, energy, average reaction time, and some structure-related information for the selected state. We also implemented a zooming feature. Since our data is generated in a way that states with similar structures are closed together in latent space, using the zoom functionality users can view states without occlusion. The right interface is a scrollable bar chart. The trajectory space pane is a bar chart. Each line mark encodes a trajectory, with the horizontal position encoding the total reaction time, that is, the longer the length of the line mark, the longer the reaction time of the trajectory. A tooltip with the selected trajectory's ID and time is implemented, which can be displayed by hovering over the line mark. We also designed a sorting

feature, by which users can sort the bar chart based on the ID number or the total reaction time of trajectories. This view can help users with task **T-S1**, **T-S2** and **T-T1**, **T-T2**.

Coarse-grained hexbin pane Fig. 4 shows the coarse-grained hexbin pane. Each hexbin is an aggregation of nearby states, and the opacity of the filled color encodes the number of states contained in the hexbin. Hovering over a hexbin can display a tooltip with its information on average energy and the number of states. There is also a text box designed on the right side to display hexbins' information when selecting a trajectory. This view can help users with task **T-S3** and **T-T4**, **T-T5**.

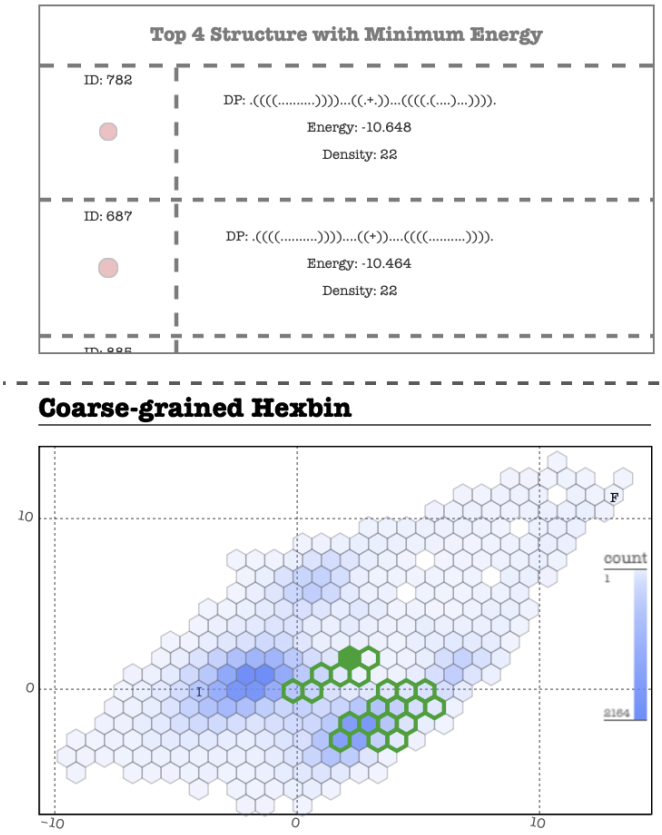


Fig. 4. The coarse-grained hexbin pane.

Energy flow pane Fig. 5 shows the energy flow pane. In this line chart, the x-axis refers to the percentage of total reaction time, and the y-axis refers to the value of energy or occupancy density. Different colored lines represent different trajectories. We designed zooming with a tooltip feature in the line chart. Users can select a line segment within a certain time period to view the energy distribution in this period (the histogram), the number of hops, and the cumulative reaction time until the selected period. This view can help users with task **T-T3**, **T-T4**.

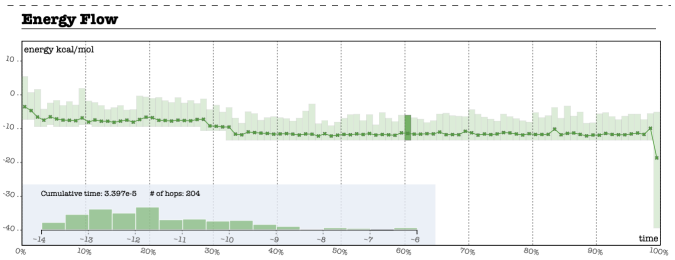


Fig. 5. The energy flow pane.

Interactions For the state space pane with a scatter chart, users can click on the point marks to show the tooltip of the most recently clicked state. Moving the slider will filter out the states based on occupancy densities and display post-filtered states. Pressing "R" on the keyboard will reset the density filter and thus show all states. For the trajectory space pane with a bar chart, clicking on at most three line marks can select three trajectories for further interactions and visualization. Selected trajectories will be shown on the scatter chart and displayed their flows in the flow view. For the energy flow pane, when only one trajectory is selected, hovering over the line mark can display a tooltip with information on the hovered percent of the trajectory; clicking a mark will stroke the bins in the hexbin pane in which states of the selected percent are contained. For the coarse-grained hexbin pane, the text box displays the initial and final states by default. When stroked hexbins exist, clicking on such hexbin enables the text box to display energy-sorted states that fall inside the hexbin.

6.2 Implementation

Manipulating the states Users can showcase latent space with a scatter plot in the state space pane. They can click the circle mark in the scatter plot to display the state information. To get a close look at states of interest and their neighbors, users can select an area and zoom it in to rescale the entangled states that are close together so that they no longer overlap with each other. Meanwhile, users can click any circle mark and show its actual secondary structure information. Additionally, if users are interested in some important states, that is the states with high occupancy density, they can slide the density slider to display the reduced state space. Furthermore, if users would like to show coarse-grained state space, they can look at the coarse-grained hexbin pane (Fig. 4) in which they can mouse over each hexbin to see the aggregated information.

Manipulating the trajectories Users can showcase up to three trajectories simultaneously in state space by clicking the line marks on the scrollable bar chart and then comparing spatial shapes of different selected trajectories. In the meantime, the energy or occupancy flows of the selected trajectories are also shown in the energy flow pane (Fig. 5) to help users to make the comparison of the energy or occupancy density change over time of different trajectories. One of the hardest problems that people would like to solve is to identify the kinetic traps in a given reaction. With i-ViDa, we provide two ways to identify kinetic traps: (i) users can visually locate dense clusters of line segments in the state space pane (Fig. 3), then zoom in and click each circle mark to gain related secondary structure information in the tooltip box. Compared with these states users may roughly determine whether a state is a kinetic trap; (ii) users can first check the energy flow pane and select flat portions in the line mark (Fig. 5) while corresponding hexbins will be highlighted synchronically. Then they can click each hexbin and a set of energy-based sorted states that belong to the bin are displayed (Fig. 4), therefore, users can easily find the lowest energy state and retrieve its secondary structure information to verdict whether the state is a kinetic trap.

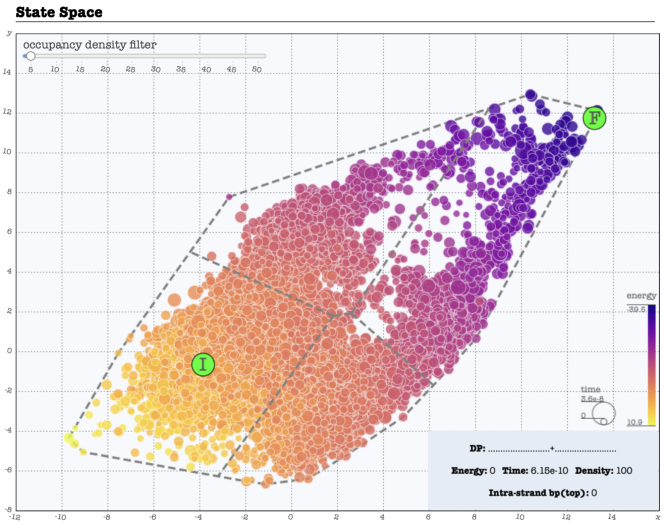
6.3 Scenario

Kevin is a graduate student working on simulating DNA reactions with the Multistrand simulator. He used ViDa to embed 100 output simulated trajectories from Multistrand into 2D space, i.e. he has the

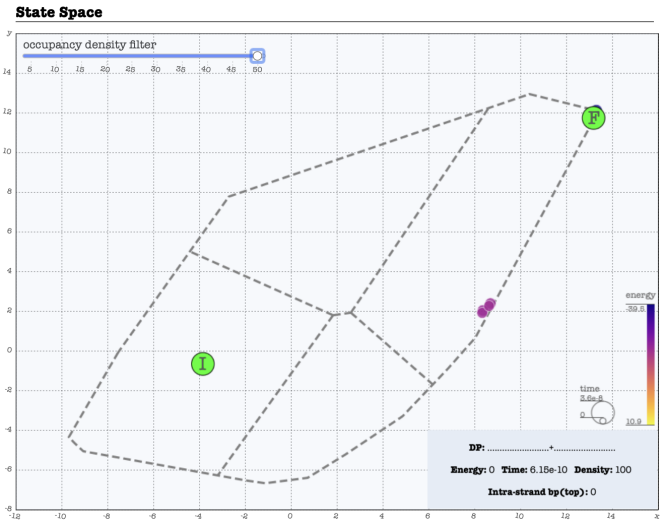
X and Y coordinates of states (secondary structures), the trajectories, and their related information. Now he would like to use i-ViDa to visualize the state and trajectory spaces in 2D to gain more insight into the underlying reaction.

6.3.1 Scenario in states

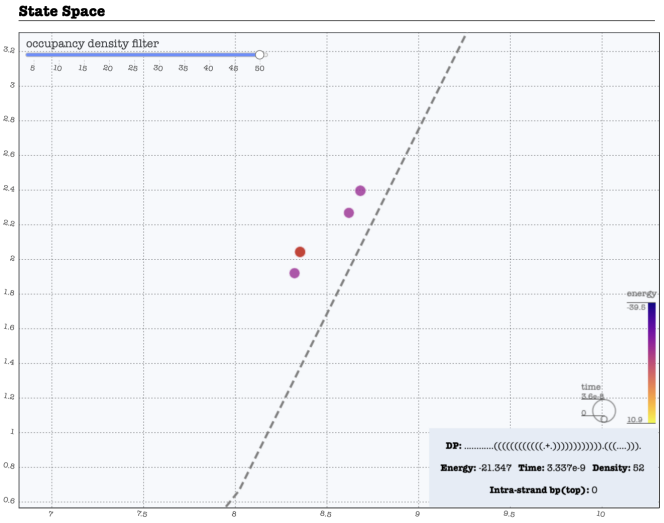
1. Kevin uploads a DNA reaction dataset into i-ViDa to visualize the state space.

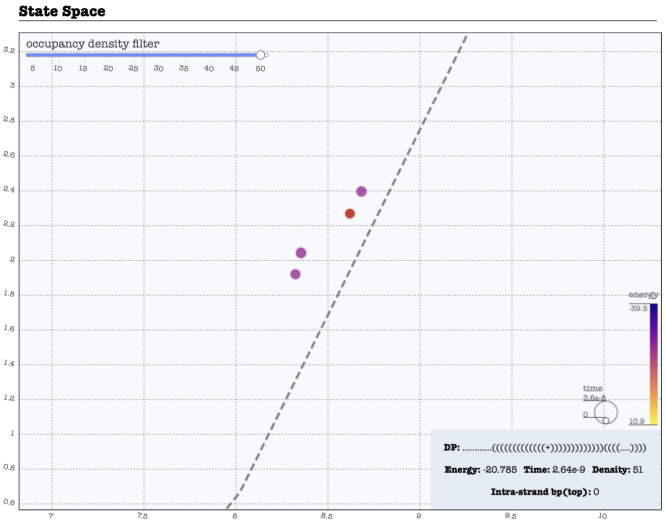


2. Kevin is unable to view the occluded and dense states shown in the state space pane. He decides to drag the occupancy density filter slider to filter out states with low densities.



3. After filtering, Kevin finds there are only a few states left in a small region that are close together. So he zooms in on the region and clicks each state to view the information in the tooltip box. As the info shows, he finds that the states mainly contain 3 or 4-stem hairpin structures formed at the end of a strand. The 3-stem hairpin has a shared stem structure with the 4-stem one, which indicates that it is in an intermediate state when forming the final 4-stem hairpin.

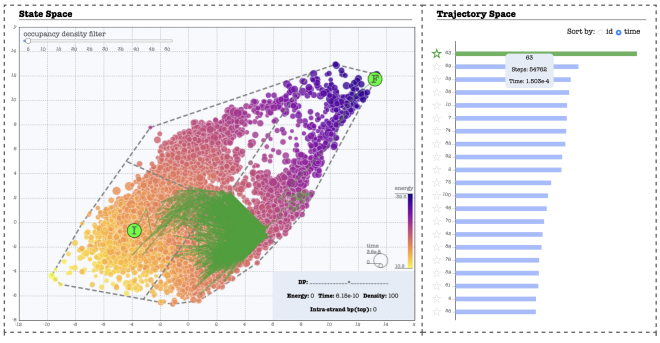




4. These findings tell Kevin that the 3 and 4-stem hairpins are dominant in the 100 simulated reactions, which may guide Kevin to study the reaction steps and mechanism.

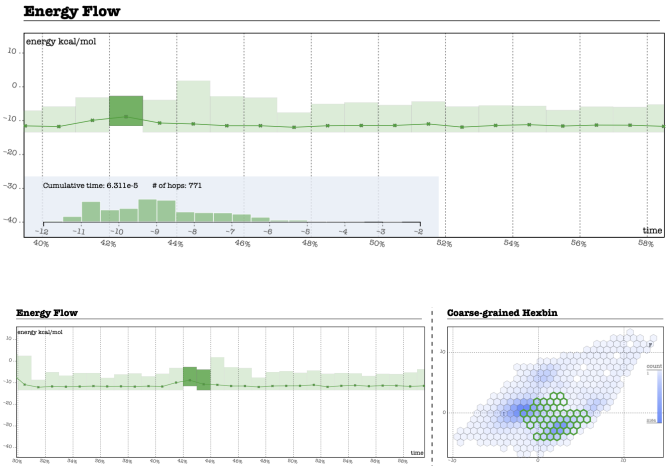
6.3.2 Scenario in trajectories

1. After seeing the state space, Kevin is more interested in the trajectory taking the largest reaction time. Therefore he first presses "R" on the keyboard to reset the filter, then clicks the sort time button as well as selects the first line mark which refers to the longest trajectory with id #63 to visualize.

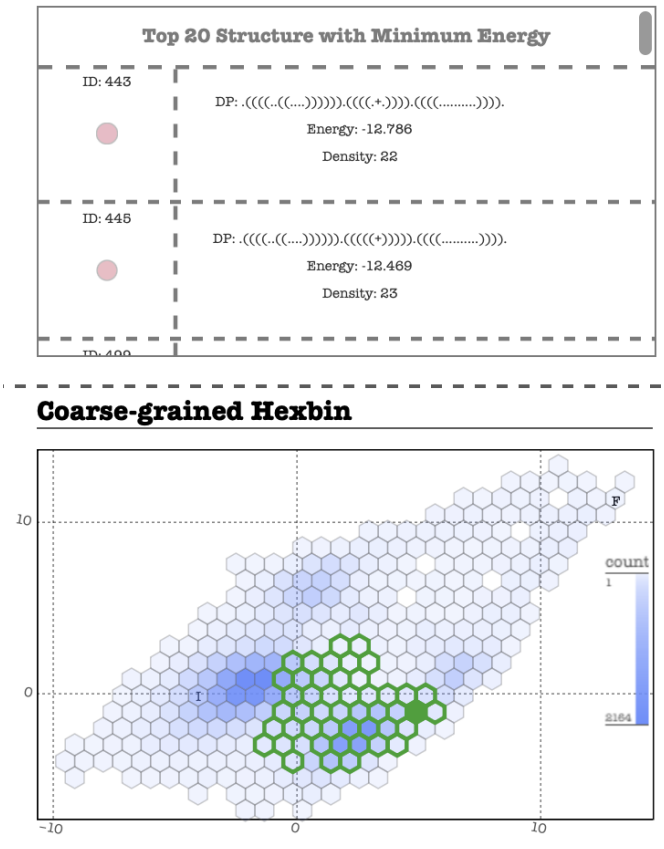


2. Kevin then hovers over the energy flow pane to view a histogram of energy and involved states in the specified period of the reaction #63.

3. Kevin finds there is an energy drop between the time period of 42% and 43%, so he suspects that there may be a kinetic trap among these states. Then he clicks these two bars and a few hexbins are highlighted in the coarse-grained hexbin pane.



4. Referring to the state space, Kevin suspects there might be a kinetic trap existing in the hexbin. He clicks this hexbin and a pop-up window presents. He views the top 20 states that fall into this selected hexbin in the pop-up, where states are ordered in ascending energies. He finds that the top several states all have the same 4-stem hairpin structure. Owing to the topmost one having the minimum local energy, Kevin attributes this structure as a kinetic trap.



5. With the implementation of i-ViDa, Kevin is able to roughly find some kinetic traps. With the knowledge of the traps' structure and energy, he can design novel DNA reactions and manipulate them, to impede or accelerate.

7 Limitation

The biggest challenge of our visualization tool is that it is very hard to track each transition in extremely long trajectories in the state space. As shown in Fig 3, the trajectory laid on the state space looks very dense and users are limited to gaining detailed information from this plot. Although we designed the density filter to filter out the low-density states, it is possible to miss some low-density states but of interest. We need to find other more accurate ways to better showcase these trajectories.

Additionally, we found that our approach cannot precisely find the kinetic traps. i-ViDa can help users narrow down the range of potential kinetic traps, but we still need some numerical analysis to eventually determine the traps with their specific structures.

Our current visualization tool is constrained by some specific DNA reactions, owing to the limitation of the dataset. To make the tool more general, we need to generate more datasets from different DNA reactions.

8 Future Work

In future work we plan to add a new feature related to K-means coarse-grained state and trajectory spaces. To better summarize the different trajectories, we would like to group them based on the major regions that those trajectories pass by. More specifically, we would like to apply K-means to separate the states into four classes, labeled as "0", "1", "2", and "3", based on the Euclidean distance between states in latent space, that is, the states are in the same cluster have very similar secondary structures and corresponding energies. The proposed idiom is shown in Fig. 6. In this pane, the original scatter plots are split into four polygons (enclosed by the K-means cluster edges with hull edges), states in each of which have similar secondary structures and energies. The asterisks refer to the centroid of each polygon, and the initial and final are labeled *I* and *F* in green circles. The colored curve encodes a trajectory. It is worth noting that when plotting trajectories, two adjacent polygons are connected by a line passing through the centroid of each polygon, therefore, users can use view the spatial shapes of trajectories without dense line-segment crossings. This view can help users with task **T-S3** and **T-T5**.

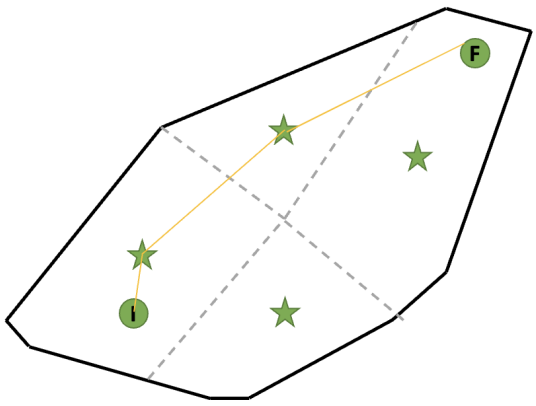


Fig. 6. The proposed K-means pane (has not yet been added in our design).

Additionally, in future 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.

9 Conclusion

In conclusion, we designed an interactive plotting tool, i-ViDa, for showing DNA reaction state and trajectory spaces. With i-ViDa, users can more intuitively compare different trajectories based on their spatial shape and position laid on the state space and energy information. Moreover, our tool can help users identify kinetic traps and reaction pathways because of the designed interactions crossing different panels. We believe our interactive plotting tool can help domain experts to better understand the output from the reaction simulators, and further help synthetic biologists and molecular programmers design novel DNA reactions with more potential applications.

10 MILESTONE

Table 3. Milestones

Milestones	Owner	Est. Time	Act. Time	Date
Introduce project background knowledge, dataset discussion, and visualization methods discussion	both	3hrs	5hrs	Sep. 21
Project pitch preparation	both	3hrs	4hrs	Sep. 27
Generate and translate datasets into certain formats	Chenwei	5hrs	5hrs	Sep. 30
Discuss all possible project tasks	both	6hrs	10hrs	Oct. 5
Discuss potentially designed prototypes	both	5hrs	5hrs	Oct. 17
Connect with domain experts to discuss the proposal	Chenwei	3hrs	3hrs	Oct. 17
Proposal writeup	both	6hrs	10hrs	Oct. 21
Provide ideas and layout for the interactive tool design	Chenwei	3hrs	3hrs	Oct. 24
Generate and design Voronoi region plots	Chenwe	6hrs	3hrs	Oct. 25
Complete interaction functionality of overview-view	Yibo	10hrs	10hrs	Nov. 1
Complete basic functionality of hexbin chart	Yibo	8hrs	8hrs	Nov. 2
Complete basic functionality of flow chart	Yibo	8hrs	8hrs	Nov. 5
Finalize interaction activities to implement	both	8hrs	8hrs	Nov. 6
Complete interaction functionality of flow chart	Yibo	10hrs	10hrs	Nov. 10
Debug and give feedback on Yibo's prototype design and discussion	Chenwei	2hrs	2hrs	Nov. 12
Rephrase the whole proposal for update writeup	Chenwei	15hrs	15hrs	Nov. 14
Discuss feedback from peer review, finalize required functionality	both	3hrs	3hrs	Nov. 15
Complete all functionality of all views	Yibo	10hrs	10hrs	Nov. 16
Discuss styling choices	both	5hrs	2hrs	Nov. 18
Debugging the final version of the tool	Chenwei	10hrs	3hrs	Nov. 25
Prepare the final presentation	both	10hrs	5hrs	Dec. 14
Document all results and write the final report	Chenwei	20hrs	15hrs	Dec. 16

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] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

[5] 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.

[6] K. R. Moon, D. van Dijk, Z. Wang, S. Gigante, D. B. Burkhardt, W. S. Chen, K. Yim, A. v. d. Elzen, M. J. Hirn, R. R. Coifman, et al. Visualizing structure and transitions in high-dimensional biological data. *Nature biotechnology*, 37(12):1482–1492, 2019.

[7] 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.