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

Chenwei Zhang and Yibo Jiao

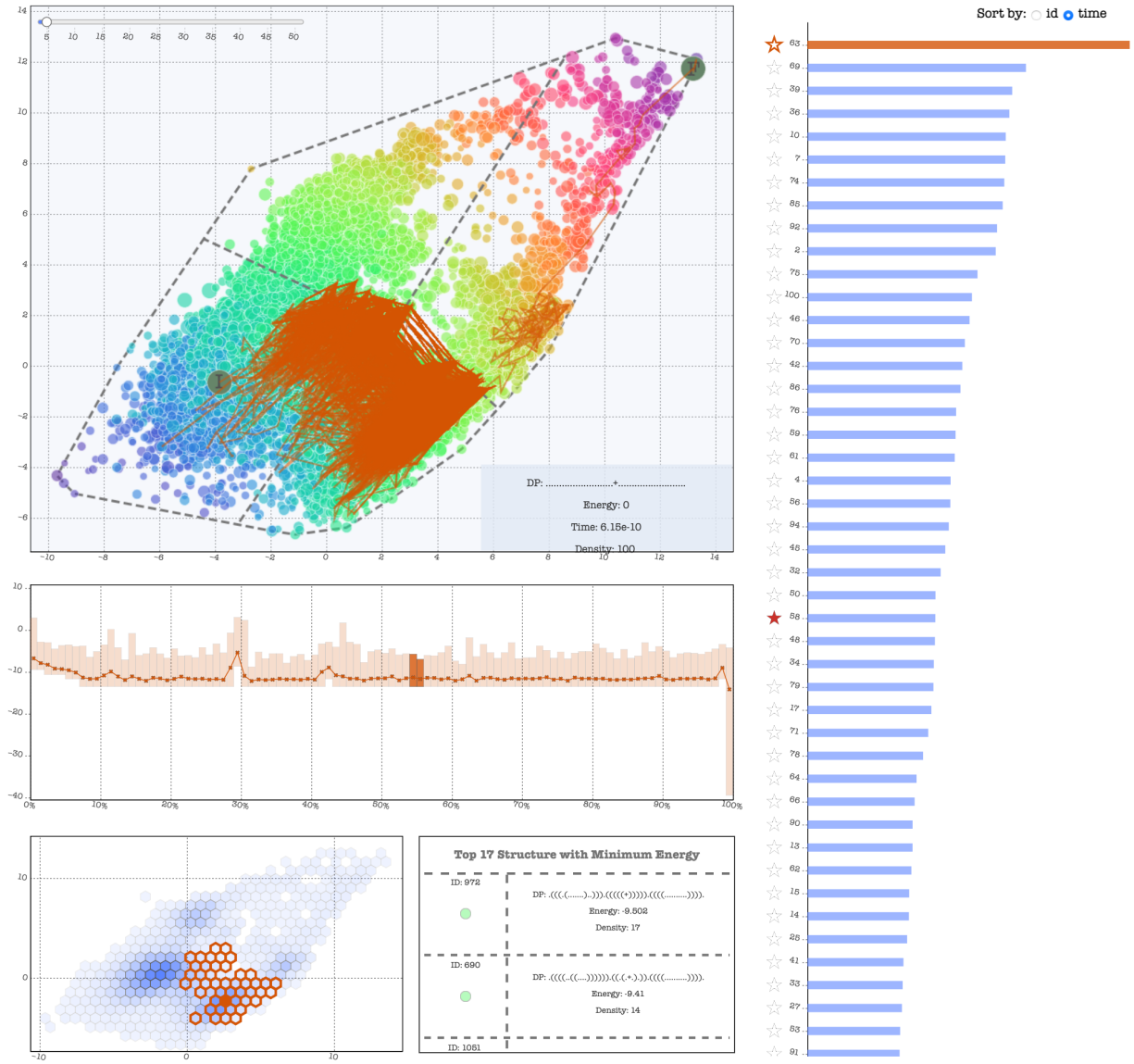

Fig. 1. Interface of i-ViDa.

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, for

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

which one unpaired DNA strand hybridizes with its complementary strand to form a fully paired duplex. In a specific DNA hybridization reaction, if there exists 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).

This project is an extension of Chenwei's RPE project, for which he designed an 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 for 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 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 word, 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 do 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 do 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 are 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 (state space) and the trajectory space.

Reaction mechanism 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.

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.

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

Reaction trajectory 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.

3 Related Work

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

4 Previous System: ViDa

This course project builds on a visualization model, ViDa, proposed in Chenwei's 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 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.

Fig. 2. The framework of ViDa.

5 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 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 dp-notation secondary structure accompanied with 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.

Since the state space is fairly large, how can we find a meaningful way to reduce the size of the state space 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 (see Eq. 3), while the rest of 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.

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, 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 strands, the number of correctly bound inter-strand base pairs, the total number of inter-strand base pairs, 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. We applied K-means to separate our dataset into four classes, labelled 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 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}
K-means class	categorical	{0, 1, 2, 3}

Table 2. Data abstraction for trajectory information.

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

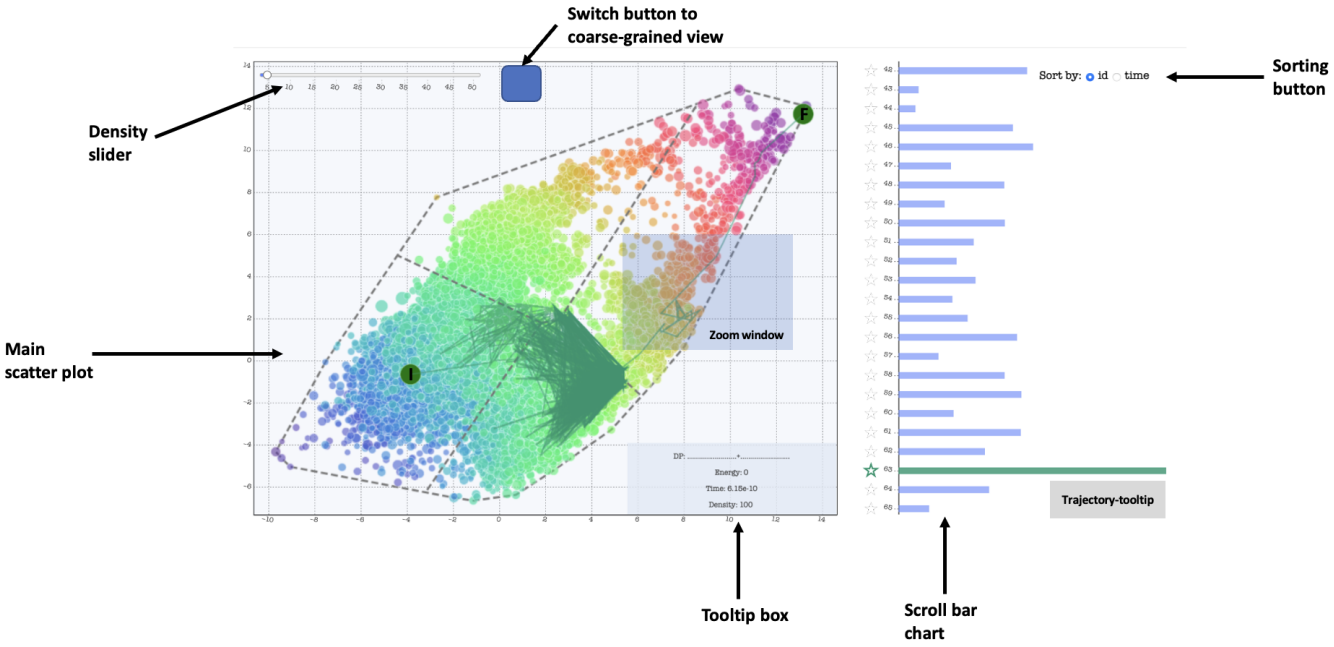

Fig. 3. Main view.

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

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

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

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 Show 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 and/or occupancy density 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

6.1 Implementation

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

6.2 Idiom Choices

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

Main view Fig. 3 shows the main view, which has two interfaces. The left interface 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. Beside the slider, a button is placed to switch the original fine-grained scatter plot to a K-means produced coarse-grained plot (Fig. 4). 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 scroll bar chart. In this 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 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**.

K-means view Fig. 4 shows the K-means view with four centroids of the coarse-grained state space. In this view, the original scatter plot are split into four polygons (enclosed by the K-means cluster edges with

hull edges), states in each of which have similar secondary structure and energies. The asterisks refer to the centroid of each polygon, and the initial and final are labelled *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**.

Hexbin view Fig. 5 shows the hexbin view. 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 of average energy and the number of states. There is also a text box designed in the right side to display hexbins' information when selected a trajectory. This view can help users with task **T-S3** and **T-T4**.

Flow view Fig. 6 shows the energy and/or occupancy density flow view. In this line chart, the x-axis refers to the percentage of total reaction time, and the the y-axis refers to the value of energy or occupancy density. Different colored lines represents different trajectories. We designed a zooming with tooltip feature in the line chart. Users can select a line segment within certain time period to view the energy range, the number of hops, and cumulative reaction time until the selected period. This view can help users with task **T-T3**, **T-T4**.

Interactions For the main view scatter chart, users can click on the point marks to show the tooltip of the most recent clicked state. Moving the slider will filter out the states based on occupancy densities and display post-filtered states. Pressing "R" on keyboard will reset the density filter thus show all states. For the main view 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 display their flows in the flow view. For the flow view, when only one trajectory is selected, hovering over the line mark can display a tooltip with information of the hovered percent of the trajectory; clicking a mark will stroke the bins in the hexbin view in which states of the selected percent are contained. For the hexbin view, 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.

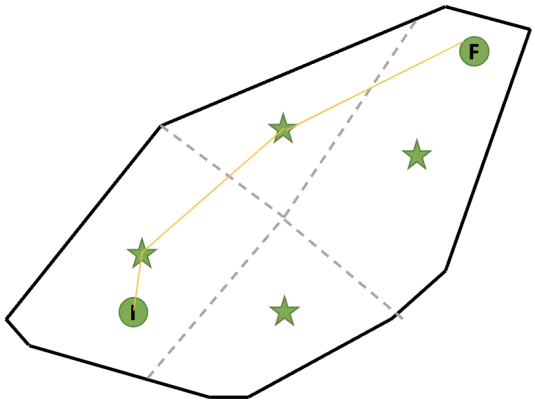

Fig. 4. K-means view.

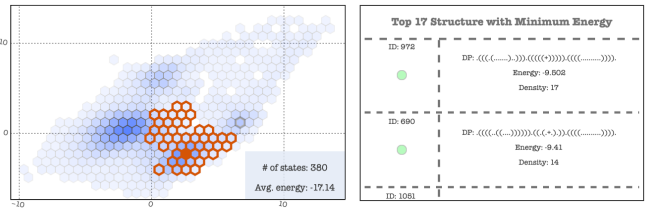

Fig. 5. Hexbin view.

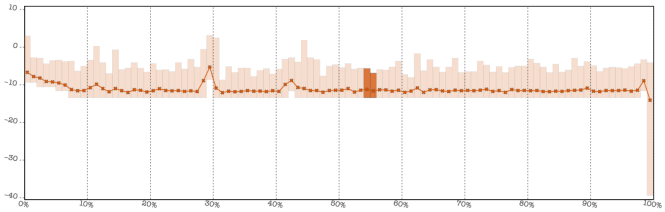

Fig. 6. Flow view.

6.3 Scenario

We propose our potential solutions by introducing scenarios of use aligned with each corresponding task discussed in Section 5.2. We deduce two scenarios w.r.t state space and trajectory space.

6.3.1 Scenario in state space

Users can showcase latent space in a scatter plot in the main view. They can click the circle mark in the scatter plot to display the state information. To get a close look of states of interest and their neighbours, users can select an area and zoom it in to rescale the entangled states that are close together so that they are 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 have two options: (i) click the top button the main view to switch to the K-means made coarse-grained scatter plot (Fig. 4); (ii) check the hexbin view (Fig. 5) in which they can mouse over each hexbin to see the aggregated information.

6.3.2 Scenario in trajectory space

Users can showcase up to three trajectories simultaneously in state space by clicking the line marks on the scroll bar chart and then compare spatial shapes of different selected trajectories. In the meantime, the energy or occupancy flows of the selected trajectories are also shown in the flow view (Fig. 6) to help users to make the comparison of the energy or occupancy density change over time of different trajectories. Users can also showcase the trajectories in the coarse-grained state space, as shown in Fig. 4. By this means, users can summarize the major reaction pathways based on the coarse-grained trajectory space. 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 main view (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 view and select flat portions in the line mark (Fig. 6) 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. 5), therefore, users can easily find the lowest energy state and retrieve its secondary structure information to verdict whether the state is a kinetic trap.

7 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.
  - (6 hours, Chenwei) Generate and design Voronoi region plots.
  - (10 hours, Yibo) Complete interaction functionality of overview-view.
  - (8 hours, Yibo) Complete basic functionality of hexbin chart.
  - (8 hours, Yibo) Complete basic functionality of flow chart.
  - (8 hours, both) Finalize interaction activities to implement.
  - (10 hours, Yibo) Complete interaction functionality of flow chart.
  - (2 hours, Chenwei) Debug and give feedback to Yibo's prototype design and discussion.
  - (15 hours, Chenwei) Rephrase the whole proposal.
- 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, Chenwei) Style, debug, and deploy the final version of visualization.
- Finalization, Dec.7 - Dec.16
  - (10 hours, both) Prepare the final presentation.
  - (30 hours, Chenwei) Document all results and write the final report.

8 DISCUSSION

9 FUTURE WORK

10 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] 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.