Data Preparation for Genomic Visualization

This documentation provides step-by-step guidance on preparing Hi-C data for use in the multi-view visualization dashboard. It outlines data preprocessing, data transformation, and the generation of 2D and 3D graph coordinates, along with edge data processing for 2D, 3D, heatmap and parallel plot visualizations. For a detailed walkthrough and all the codes for data processing, visit: https://github.com/Devopriya-Tirtho/HiC-Data-Processing-for-Multi-view-Visualization-Dashboard

1. Data Preprocessing for Visualization

Input Data Description: Hi-C Contact Matrices

Hi-C data consists of matrices that capture interaction frequencies between pairs of genomic bins. These matrices are essential for understanding the spatial organization of a genome. Each entry in the matrix represents an interaction value between two genomic bins. The raw Hi-C data is typically in tab-separated, comma-separated, or space-separated format. Preprocessing is required to make the data compatible with visualization tools.

Step 1: Initial Data Cleaning

- Goal: Format the raw Hi-C data by replacing whitespace characters with commas to standardize the data structure.
- Action: Use a text editor or a script to replace all whitespace with commas. Ensure that the data remains in a structured and readable format.

2. Data Transformation Process

Overview

The cleaned Hi-C data must be converted into an adjacency matrix format for further analysis and visualization. This involves several transformation steps:

Step 2: Transformation to Adjacency Matrix

1. Script 1: Use a Python script to transform the formatted Hi-C data into an adjacency matrix. This matrix will have rows and columns representing genomic bins, with interaction frequencies as the matrix values.

2. Script 2: Convert the adjacency matrix from CSV format to TXT while ensuring no consecutive bins are lost. This preserves the data's integrity.

3. Script 3: Convert the TXT file into a JSON format optimized for efficient visualization. The JSON file will be used as input for the visualization system. Output: An adjacency matrix and a JSON file suitable for use in the visualization dashboard.

3. Data Processing for Visualization Methods

3.1 Extraction of 3D Graph Coordinates

Algorithm: 3D Force-Directed Layout

- Description: The 3D graph layout algorithm models nodes (genomic bins) and edges (interactions) as a physical system. Forces act on the nodes to spread them in a balanced 3D configuration.
- Forces Involved:
  1. Repulsive Force: Prevents nodes from overlapping, ensuring even distribution.
  2. Attractive Force: Draws connected nodes, reflecting the strength of interactions.

Steps to Generate 3D Coordinates

1. Run the 3D Graph Layout: Use a 3D force-directed graph library to simulate the forces acting on nodes.

2. Stabilize the Layout: The simulation runs iteratively until the node positions stabilize.

3. Export Coordinates: Once stabilized, export the x, y, and z coordinates of each node to a CSV file.

Preparing the Data

1. Smoothing: Apply a smoothing algorithm to reduce jitter and refine node positions.

2. Conversion: Convert the smoothed coordinates from CSV to JSON format for the visualization dashboard.

3.2 Extraction of 2D Graph Coordinates

Algorithm: Modified ForceAtlas2

- Description: The ForceAtlas2 algorithm is adapted to handle weighted graph data, positioning nodes in 2D space based on interaction strengths.
- Steps:
  1. Run the Algorithm: Execute the 2D layout algorithm to generate node positions.
  2. Export Coordinates: Extract the 2D coordinates and store them in a CSV file.

Preparing the Data: Convert the CSV file to JSON format, ensuring compatibility with the visualization system.

4. Heatmap Data Preparation

Steps to Prepare Data for Heatmap Visualization

1. Load the Hi-C Data: Use a data processing library to load the data into a structured format.

2. Sort the Data: Organize the data by source and target bins to maintain consistency.

3. Create a Symmetric Matrix: Convert the data into a symmetric sparse matrix, ensuring bidirectional interactions are captured.

4. Export to JSON: Extract non-zero matrix entries and convert them to JSON format for heatmap visualization.

5. Edge Data Preparation for Visualizations

Overview

Different visualizations require specific subsets of the interaction data. The complete data is used for heatmaps, while significant interactions are prioritized for 2D, 3D graphs, and parallel plots.

Steps for Edge Data Preparation

1. Load Processed Hi-C Data: Use a data processing tool to load the cleaned and formatted data.

2. Select Significant Interactions:

- Top 20 Edges: For each node, select the top 20 interactions based on weight.
- Top X% Edges: Additionally, select the top X% of all interactions across the dataset.

3. Compile Edge Data: Create a DataFrame with source, target, weight, and interaction type.
4. Export to JSON: Convert the DataFrame to JSON format for use in 2D, 3D, and parallel plot visualizations.

Conclusion

Follow this documentation to transform and prepare Hi-C data for genomic visualization. The process involves cleaning, transforming, and exporting data in suitable formats for different visualization methods in the dashboard system. Ensure all steps are followed precisely to maintain data integrity and visualization accuracy.