SBOLExplorer - Process Book

Russell Kennington - u0946645 - arussellk@gmail.com
Jiahui Chen, u0980890 - jiahui.chen@utah.edu
https://github.com/arussellk/dataviscourse-pr-sbolexplorer

Overview and Motivation:

This project aims to visualize a dataset of synthetic genetic parts as well as the inheritance relationships between them. The Synthetic Biology Open Standard (SBOL - http://sbolstandard.org/) defines a data format that represents synthetic genetic building blocks, which may be combined to create genetic constructs or larger, parent synthetic genetic building blocks. SBOLExplorer will provide a way to view and navigate SBOL data to give researchers deeper understanding of building block relationships, what genetic building blocks certain genetic parts are comprised of, and more efficient discovery of their data.

Related Work:

SBOL Standard Data Format: http://sbolstandard.org/
SynBioHub: https://synbiohub.org/

A searchable repository of the SBOL genetic parts and designs comprised of different combinations of them.

SBOLGraph Library: https://github.com/udp/sbolgraph

A library that manipulates SBOL data in a graph form, we use it to get our visualization data.

Questions:

- What subparts compose the selected genetic part?
- How do these subparts work together to make this genetic part?
- What is the detailed information for this genetic part?

These questions originally motivated our visualization designs as well as dictated what visualizations we used.
After the peer review of the project, we had the additional question:

- Would be useful to be able to download a genetic part's data from the visualization?

The answer is no, so we didn't add anything to our visualization that would enable a download.

Data:

Our data is sourced from the API that SynBioHub's data is sourced from. An example data point can be found through SynBioHub (e.g. https://synbiohub.org/public/igem/BBa_K1407008/1).

There are two forms of data we will be working with:

- Search result data
- Tree data

The **search result data** can be obtained through SynBioHub, though we have already saved a sample search result for offline use. The search data looks like this:

```
[
  {
    "_id": "104450",
    "_index": "part",
    "_score": 0.0025428662,
    "_source": {
      "description": "green fluorescent protein derived from jellyfish Aequeora victoria wild-type GFP (SwissProt: P42212",
      "displayId": "BBa_E0040",
      "graph": "https://synbiohub.org/public",
      "keywords": "BBa E0040",
      "name": "GFP",
      "pagerank": 0.00031315985719595617,
    },
  },
  ...
]
```

Notice that there is a score attribute on a search result. This will be used to create the relevance bar in the search results list view.

The **tree data** is received through the https://github.com/udp/sbolgraph library, which queries the API for a specific tree and returns a graph object for each genetic part that's queried. Some data processing has been implemented to make the returned graph better for our purposes. Additionally, we have already saved sample trees for offline use. A sample tree looks like this:

```
{
  "uri": "https://synbiohub.org/public/igem/BBa_B0030/1",
  "sequence": "attaaagaggagaaa",
  "children": [],
  "range": [63, 77]
},
```

Note that a tree has a range attribute consisting of a start and end value. This range indicates where this node fits into the sequence of its parent node. The range will be used to create the subsequence indicators in our tree.

More data processing may be needed as we discover ways our visualization can be improved. We are using TypeScript and have good knowledge of the library which means we can quickly pull in data which we need.

Exploratory Data Analysis:

There's currently no visualizations of SBOL data and we did initial data analysis by consulting Michael Zhang, who's thesis project is the creation of SynBioHub. He knows the data well because he's worked with it extensively and also belongs to a research group that is heavily involved with the creation and maintenance of SBOL.

We gained crucial information about the size and scope of genetic parts which is important to visualization decisions regarding navigation and information density:

- Average number of levels of inheritance for a genetic part:

  3, maximum is 6
- Average number of child genetic parts that a parent genetic part is comprised of:

  3, maximum is 10 but when the number of child parts is high the total inheritance levels are low.

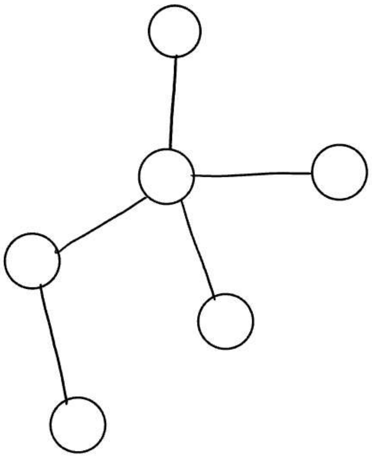
Knowing these 2 facts led us to decide to not include navigation in our visualization as the information for each genetic part will not be incredibly dense. We think that scaling the visualization will be enough to show the information for all genetic parts.

Design Evolution:

We did not deviate from the project proposal. Below is the evolution of our main visualization's design and all the additional visualizations we will include.

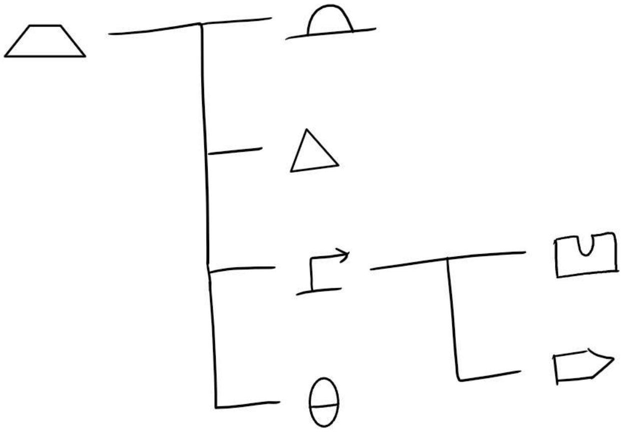
Main Visualization:

Prototype 1: Because the most important aspect of our data is the relationship between data points, we initially thought of a graph visualization for our main visualization of genetic parts where nodes represent genetic parts and edges represent inheritance relationships between them:

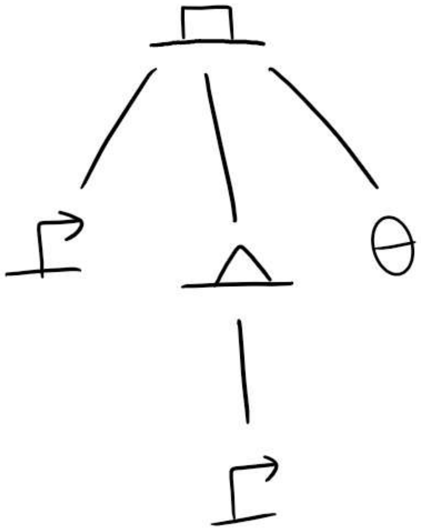


We decided a graph visualization would not explicitly show inheritance/dependency relationships, so we moved on to consider tree visualizations.

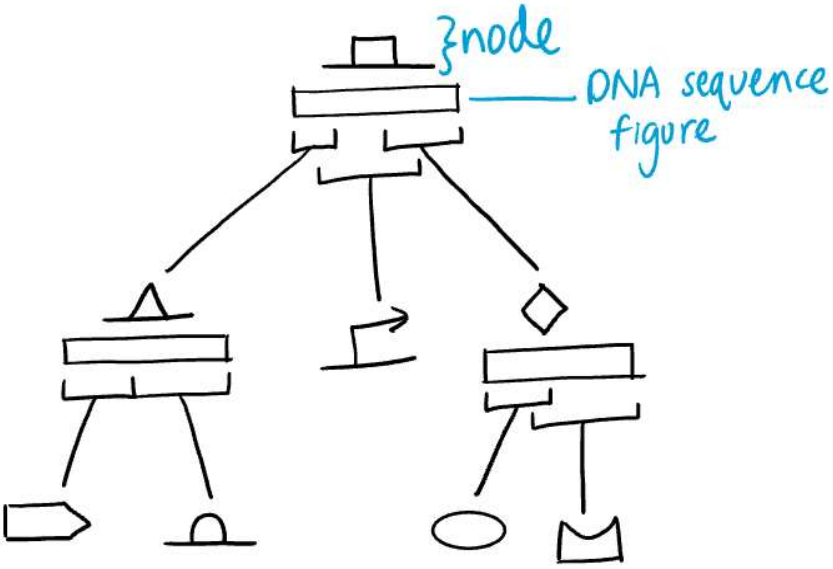
Prototype 2: Left to Right Tree - We first considered a left-to-right tree visualization to show dependencies more clearly. We also decided to use the glyphs for each node (the glyphs are part of the SBOL standard and indicate a certain type of genetic part) as the marks instead of the same shape for all nodes to show more information on first glance.



Prototype 3: Top Down Tree - We decided to do a top-down tree instead of a left-to-right tree because according to Michael a top-down tree would be more readily understood by biologists, and we want our visualization to be as useful and understandable as possible for our target audience.

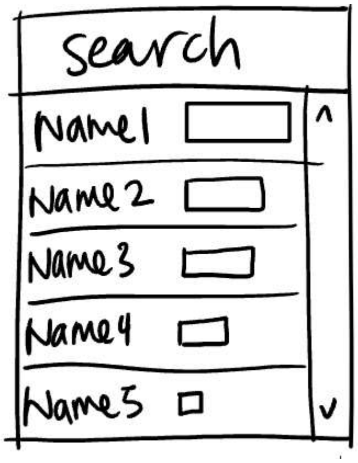


Final main visualization design: Top down tree with DNA sequence figure - this is the same as the top-down tree design, but each node also has a horizontal bar figure representing the genetic part's DNA sequence. This figure will indicate what partitions of the DNA sequence consist of its children's DNA sequences. We decided to add this DNA sequence figure because a visual representation of how the sequence dependencies are structured would be useful to researchers and isn't currently available anywhere. We thought this would make our visualization more unique and much more useful.

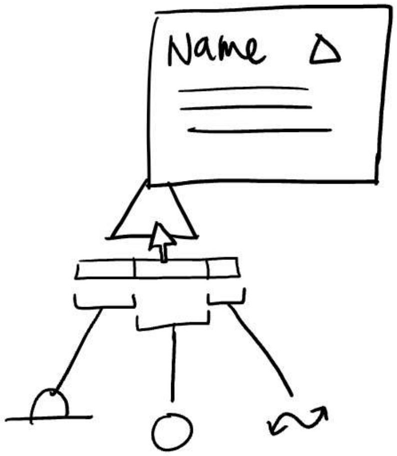


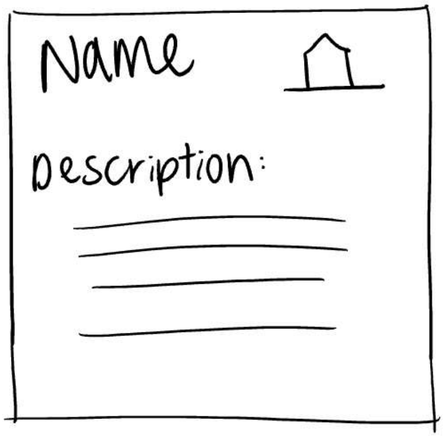
Additional Visualizations:

Search Box with Popularity Indicator - Our visualization will have a search functionality, where a string can be input and the search results will be all the SBOL parts that the query returns. Within the search results display box, there will be one one bar for each genetic part in the search results indicating its popularity (page rank, which is included in each data point). This lets users quickly discern if SBOLExplorer has found relevant results and how many of the results are worth viewing.

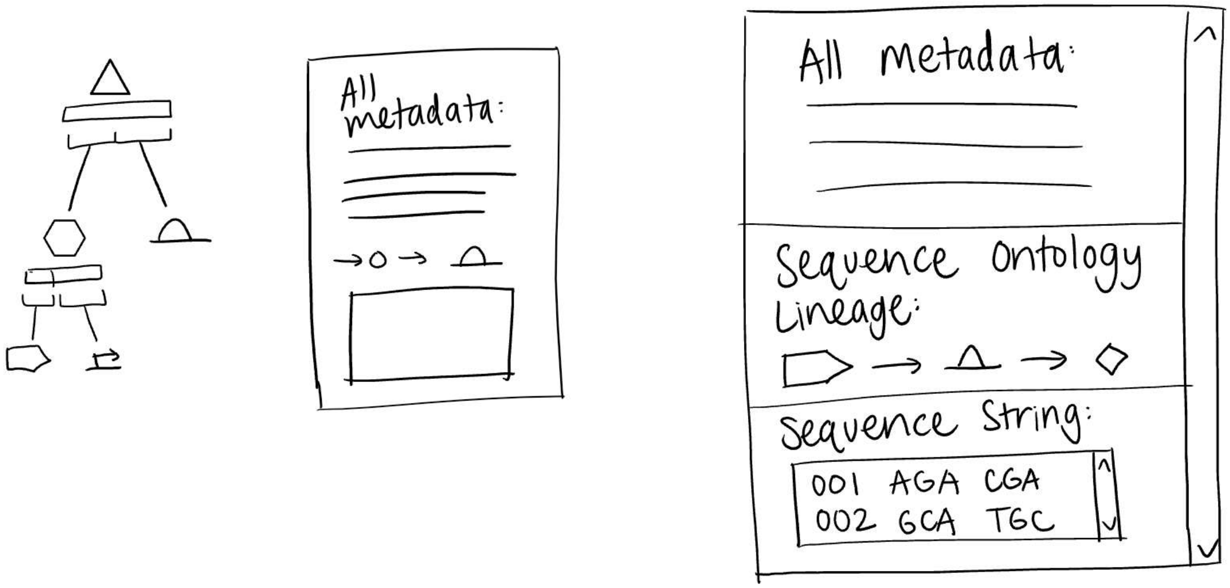


Basic Info Box - A box that displays when hovering over any node/genetic part. It will show brief information such as full name, the glyph, and version.





Thorough Info Box - This box shows all of a genetic part's metadata when the genetic part's node is clicked.



Implementation:

Currently we have implemented:

- A working project that uses an SBOL library with Webpack
- Local Search Result and Tree data as well as code that can query it from the data API

Implementation Plans:

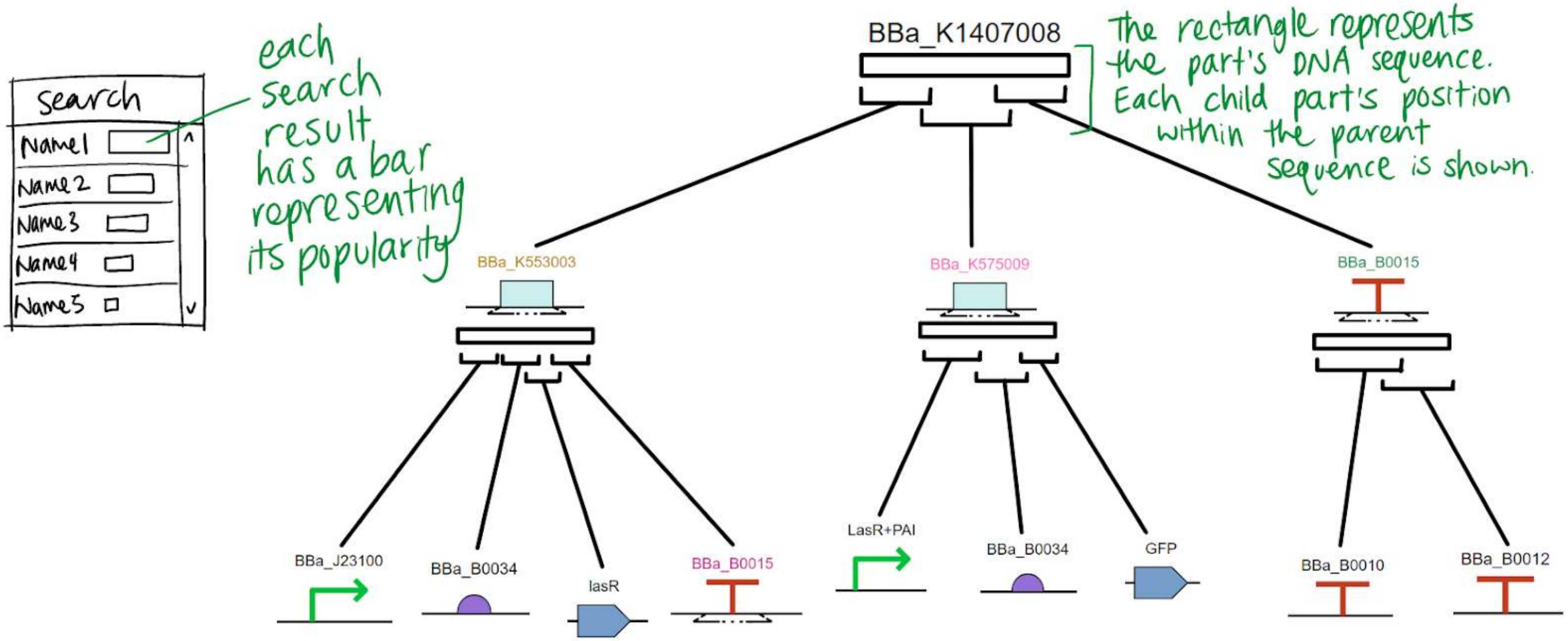
Here is a detailed mock-up of our project plans, including interaction annotations:

Overall Visualization Goal:

A tree of the data point/genetic part: https://synbiohub.org/public/igem/BBa_K1407008/1

The JSON of this data point is located at:

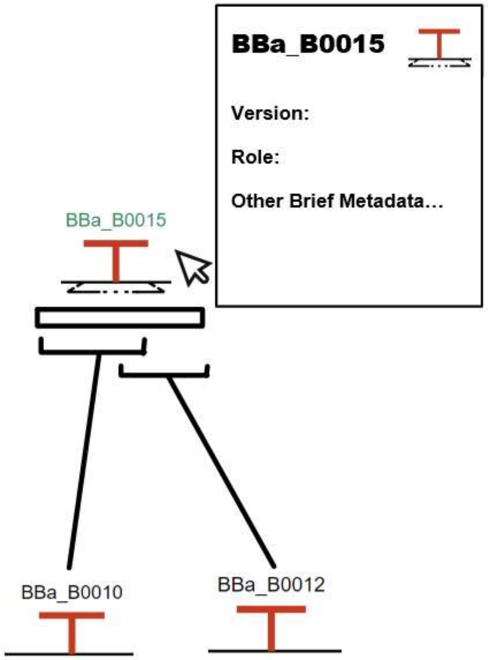
https://github.com/arussellk/dataviscourse-pr-sbolexplorer/blob/master/src/data/trees/BBa_K1407008-with-range.json



each search result has a bar representing its popularity

The rectangle represents the part's DNA sequence. Each child part's position within the parent sequence is shown.

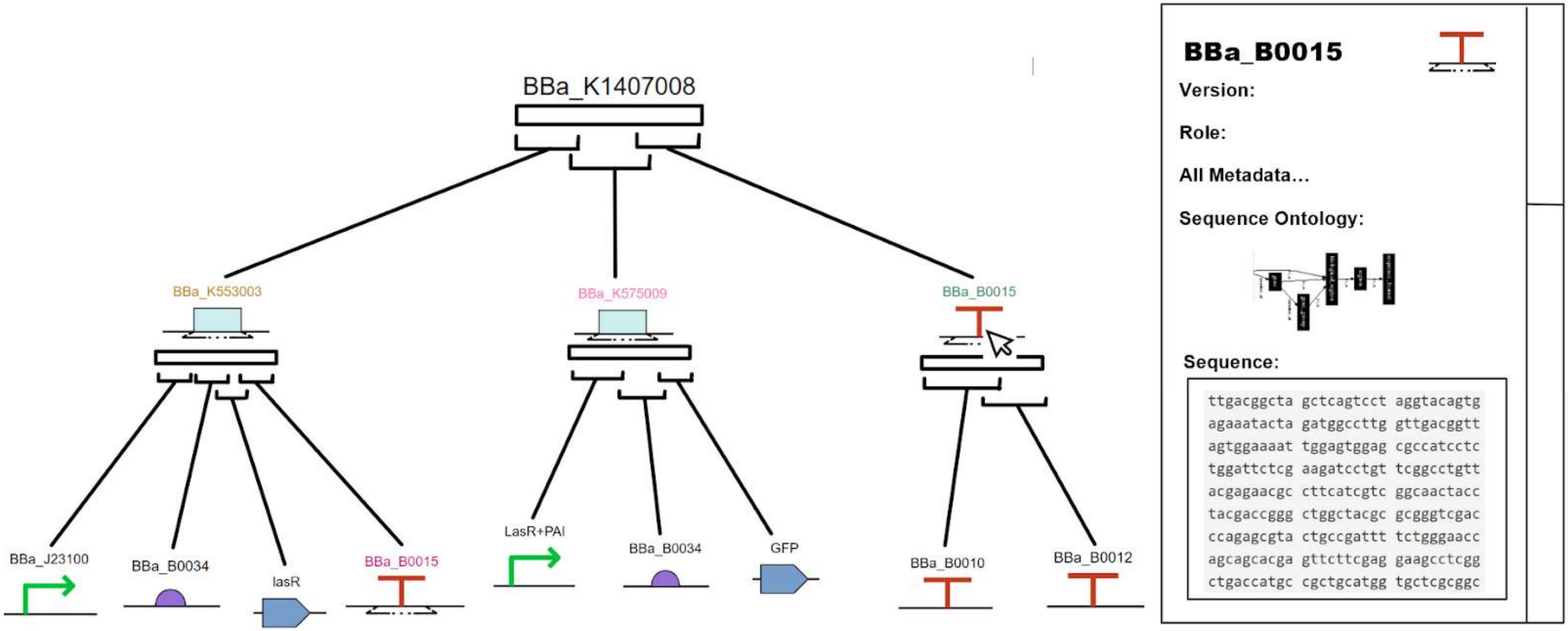
Tree Interaction: Hover



When a node in the tree is hovered over,
an infobox containing brief metadata shows up.

Tree Interaction: Click

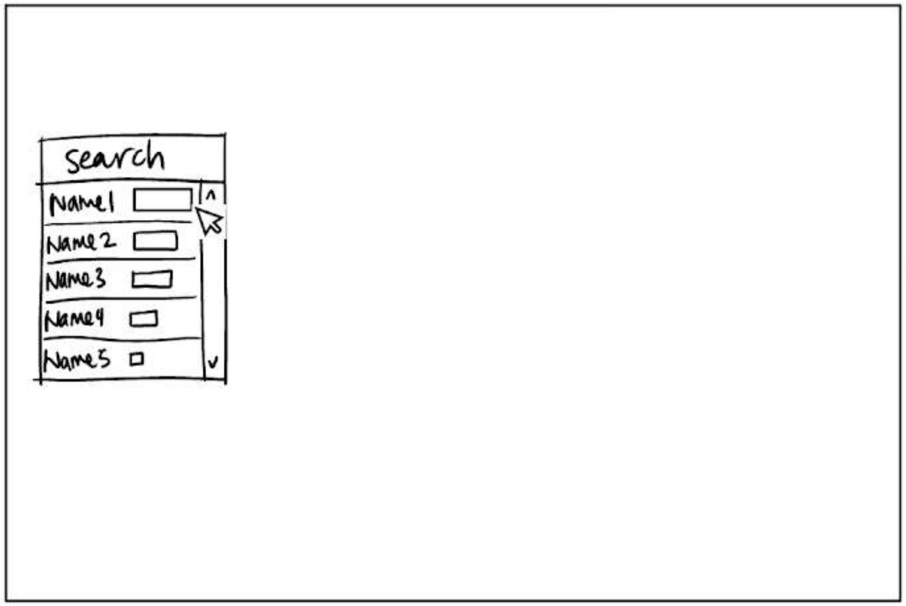
When a node in the tree is clicked on, a large, scrollable info box showing all of the genetic part's metadata appears on the page



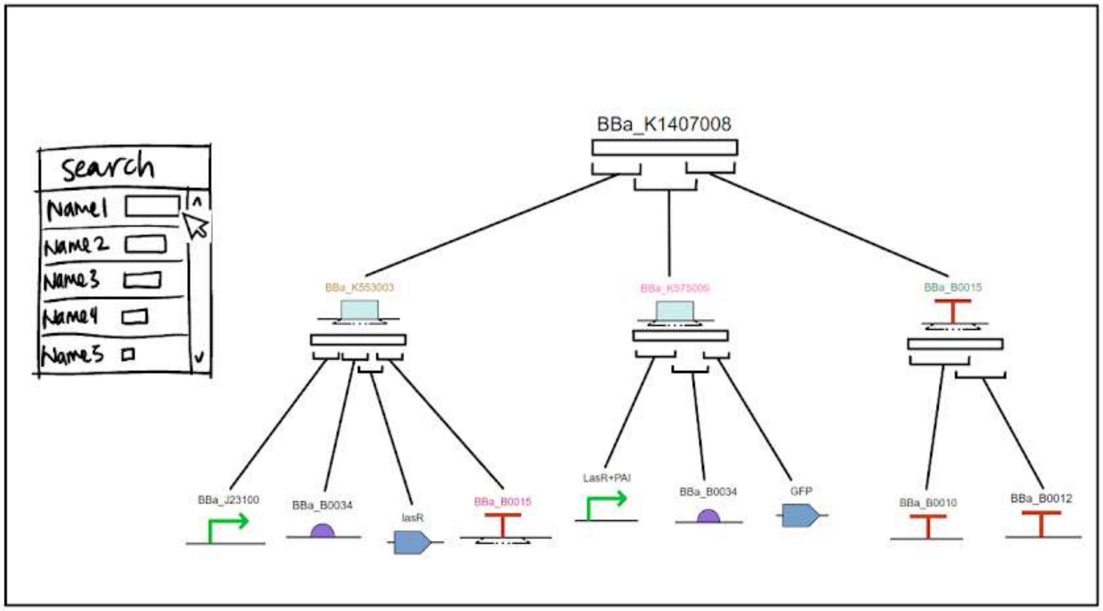
```
ttgacggcta gctcagtcct aggtacagtg
agaaatacta gatggccttg gttgacggtt
agtggaaaat tggagtggag cgccatcctc
tggattctcg aagatcctgt tcggcctgtt
acgagaacgc cttcatcgtc ggcaactacc
tacgaccggg ctggctacgc gcgggtcgac
ccagagcgta ctgccgattt tctgggaacc
agcagcacga gttcttcgag gaagcctcgg
ctgaccatgc cgctgcatgg tgctcgcggc
```

Search Interaction:

When a user clicks on a search result, the selected genetic part is passed into the sbolgraph library, which gives us the genetic part's tree data.



The tree of the selected part renders in the area to the right of the search box. If a tree is already rendered, the tree of the selected part will replace it.


search
Name1
Name2
Name3
Name4
Name5
BBa_K1407008
BBa_K553003
BBa_K575009
BBa_B0015
BBa_J23100
BBa_B0034
lasR
BBa_B0015
LasR+PAI
BBa_B0034
GFP
BBa_B0010
BBa_B0012