Materials and Methods

Basic tasks:

reading databanks and parsing into usable file

reading file, determining structures

displaying graph

(react on user input to display different structures and information)

Structure:

browser

js

d3

external scripts in Python

Functions (describe parameters)

readFile

handleContent

groepering

determineTiers

contains

matchcriteria

makeGraaf

radius

gravity

coordinates

future work:

displaying single tier

changing graph based on clicks

displaying information based on selected node

vergelijkbare tools?

Datasets

3-ecoli

7-ecoli

coli-albertii

testset?

Eerdere sets (sga, asqg)?

1. Overview. Stringit is a tool that visualises the data inside assembly graph files. The file is loaded in the webapplication, and is processed to a visual graph, which is displayed on screen. There, several tools can be used to analyse the data. The graph can be exported for use in other tools (such as alignment mappers).

2. Stringit. Stringit is a browser-based tool, where users can upload their assembly graphs, after which they can analyse the graph network in the browser window.

It does this by parsing the input file to a javascript object containing all the information on nodes and edges. Using information contained in the graph file, it groups the nodes based on the origin of the sample, creating differently coloured groups and making co-assembly analysis possible. Zooming into different tiers is possible, each consecutive one showing less overall sequence, and more detail. Based on the zooming tier, nodes get aggregated into larger nodes. This way, only information relevant for that level is displayed.

The nodes give information (via a mouse-over HUD) about the sequence(s), and edges denote overlap between nodes. When the sequence diverges because of variation, this is visually represented by two different nodes that converge behind the variation (such as a SNP or indel), which forms a ‘bubble’. The user can analyse any overlap and divergence between species or strains with this combination of zooming tiers, node aggregation, and bubble recognition.

2.1 Input data. As input for Stringit, a graph file is needed. Several assemblers are capable of producing such a file. So far, the graph formats of SGA and 454 ("Newbler") are implemented. However, any graph file is usable, in principle. The formats are recognized internally and processed into a JSON object, containing all the information of the original nodes and edges.

2.2 D3. Stringit uses a powerful javascript library to handle the graph data called d3. D3 is short for Data Driven Documents. It is used to bind data to DOM elements. Stringit uses a force-directed graph layout, one of several ways D3 can be used to display graph data. Nodes representing sequences are connected by edges representing overlap, and the internal d3.js mechanism lay this out on screen. Additional functionality is added with a mouse-over display that shows information on the current node, as well as zooming in and out to regions of interest.

3. Features:

3.1 reading databanks and parsing into usable file

3.2 reading file, determining structures

3.3 displaying graph