MSAtoGFA: a Graph Representation of Multiple Sequence Alignment

Fawaz Dabbaghie[1,2], Tobias Marschall[1], Olga Kalinina[2]

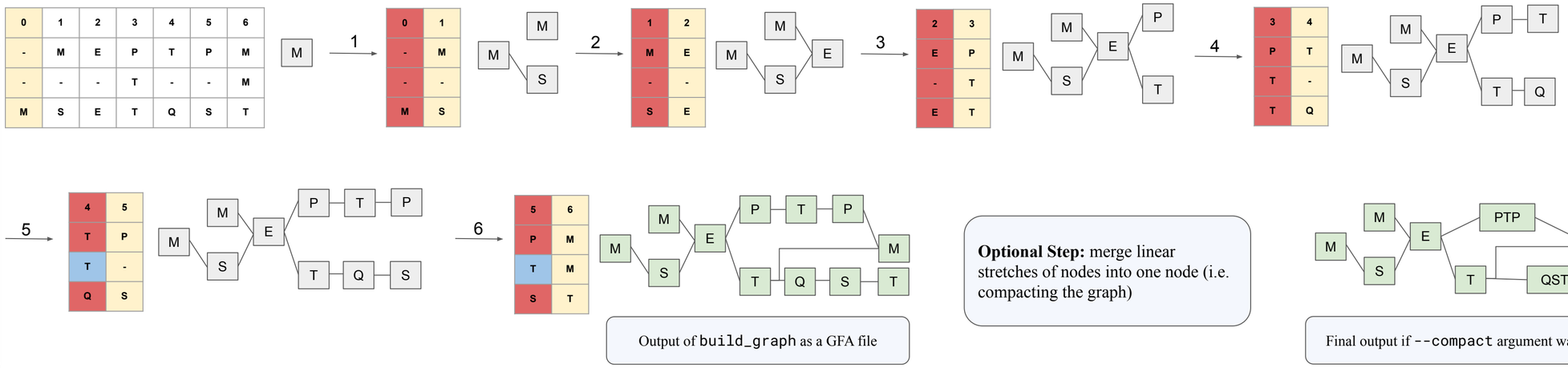
[1]Institute for Medical Biometry and Bioinformatics, University Hospital Düsseldorf, Moorenstr. 5, 40225, build. 17.11, Düsseldorf, Germany, , [2]Helmhotz Institute for Pharmaceutical Research Saarland, Campus E8 1, 66123 Saarbrücken, Germany.

Abstract: The use of variation graphs to represent the variations between many sequences has proven to be very useful and intuitive than working with many linear sequences. Therefore, many software tools have been developed for visualizing such graph data structures, e.g. Bandage [1], GFAviz [2], MoMI-G [3]. However, most of the genome graph and variation graph tools do not work with amino acid sequences of proteins, but only with nucleic acid sequences. Yet, protein sequences can still benefit from the graph data structure and the visualization abilities.

MSAtoGFA is a software tool that can be used for:

- Converting an amino acid or nucleic acid MSA (Multiple Sequence Alignment) into a Graphical Fragment Assembly graph (GFA).
- GFA files can be visualized and analyzed with many tools.
- Adds paths to the graph and help visualizing the differences between the sequences
- MSAtoGFA is written in Python and is lightweight and fast, can be used with any standard computer.

The algorithm runs in an iterative fashion, going through each column of the MSA.
In each iteration we keep track of the current column (yellow) and the previous one (red).
In each iteration nodes are generated and connected if necessary.
If previous column has a letter and current is a gap (e.g. T in 4), then that letter is pushed until we reach another letter in current and connect them (blue cells)

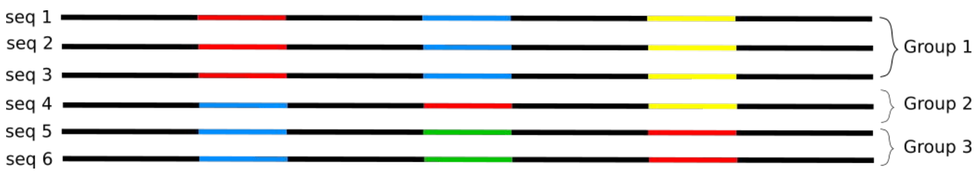


Output of `build_graph` as a GFA file

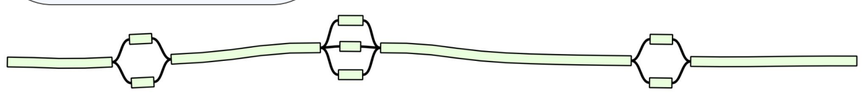
Optional Step: merge linear stretches of nodes into one node (i.e. compacting the graph)

Final output if `--compact` argument was given

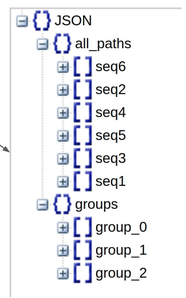
First Step: taking the 6 sequences here that have 3 heterozygous positions.
Running the command will output a graph in GFA format and the groups information as a JSON file. Same sequences are grouped together.
Where the JSON file has information to which sequences belong to which group, and the path in the graph for each group.
Graph visualized here using GFAviz




```
$ msa_to_gfa build_graph -f in_msa.fasta -o
output_graph.gfa --compact
```



`output_graph.gfa`

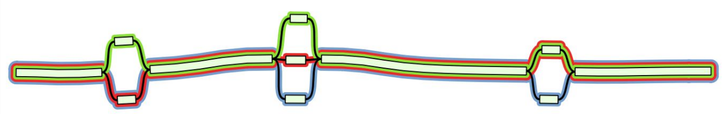


`output_graph_groups.json`

Second Step: Taking the raw graph and the JSON file from the first step, user can either choose to add all paths with `--all_groups` or `--some_groups` to add one or more group or seq.
E.g. User can choose to visualize only Group1 or only seq 6.
These paths can then be colored in GFAviz for example.

```
$ msa_to_gfa add_paths -g output_graph.gfa --in_groups
output_graph_groups.json --all_groups
```

Group 1: Green
Group 2: Red
Group 3: Blue



Time and Memory
Constructing a graph from a 9800 sequences MSA of the DNA-directed RNA polymerase subunit alpha protein (rpoA) took around 2 seconds on a standard laptop, and consumed around 120 MB of memory.
Adding all groups as paths back to the graph took around 0.3 seconds and consume 80 MB of memory.

https://bit.ly/3oobkGX

@FawazDabbaghieh

[1] Ryan R. Wick, Mark B. Schultz, Justin Zobel, Kathryn E. Holt, Bandage: interactive visualization of de novo genome assemblies, Bioinformatics, Volume 31, Issue 20, 15 October 2015, Pages 3350–3352. [2] Giorgio Gonnella, Niklas Niehus, Stefan Kurtz, GfaViz: flexible and interactive visualization of GFA sequence graphs, Bioinformatics, Volume 35, Issue 16, 15 August 2019, Pages 2853–2855. [3]Yokoyama, T.T., Sakamoto, Y., Seki, M. et al. MoMI-G: modular multi-scale integrated genome graph browser. BMC Bioinformatics 20, 548 (2019)