Supplementary Material - Sequence Alignment on Directed Graphs

Vaddadi Naga Sai Kavya Kshitij Tayal Rajgopal Srinivasan
Naveen Sivadasan*

TCS Research, Hyderabad, India - 500081
* *naveen.sivadasan@tcs.com*

1 Genome variation graph representation

Genome variation graphs are represented in various formats (Novak et al., 2017). In our work, we consider genome variation graphs that are directed graphs and with vertices labeled using variable length sequences. The reference sequences are encoded as directed walks in these graphs. Figure 1 shows the visualization of a genome variation graph with 6 vertices and 8 directed edges.

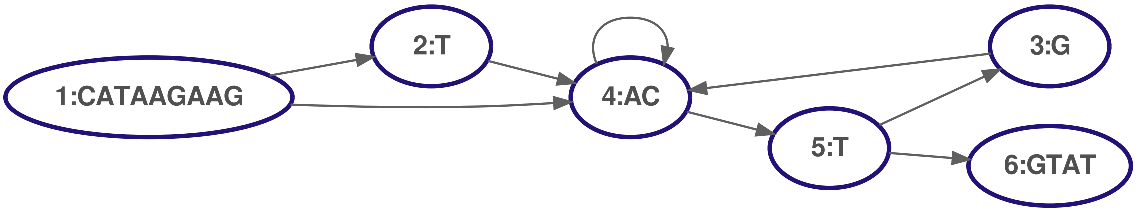

Fig 1. Genome variation graph. Vertices are labeled with variable length sequences. Girth of this graph is 2 because of the self loop in vertex 4.

1.1 Graphical Fragment Assembly (GFA) file format

V-ALIGN supports the standard GFA (Graphical Fragment Assembly) format (GFA, 2017). In this representation, the underlying graph is a directed graph where the vertices labeled using variable length sequences. The graph has an additional property that each vertex u is assumed to have two orientations, namely, the forward orientation which is denoted as $u+$ and the reverse-complemented orientation which is denoted as $u-$. The orientation information is used while specifying the directed edges. Directed edge $(u+, v-)$ indicates the traversal of vertex u in the forward direction and vertex v in the reverse-complemented direction. Hence, the sequence corresponding to edge $(u+, v-)$ is obtained by concatenating the sequence label of u and the reverse-complemented sequence label of v in the same order.

Table 1. Directed edges involving vertices u and v and their corresponding sequences

Directed edge	**Sequence**
$(u+, u+)$	ATGATG
$(u+, u-)$	ATGCAT
$(u-, u+)$	CATATG
$(u-, u-)$	CATCAT
$(v+, v+)$	ATAATA
$(v+, v-)$	ATATAT
$(v-, v+)$	TATATA
$(v-, v-)$	TATTAT
$(u+, v+)$	ATGATA
$(u+, v-)$	ATGTAT
$(u-, v+)$	CATATA
$(u-, v-)$	CATTAT
$(v+, u+)$	ATAATG
$(v+, u-)$	ATACAT
$(v-, u+)$	TATATG
$(v-, u-)$	TATCAT

Table 1 gives the possible directed edges and their corresponding sequences involving two vertices u and v with corresponding labels “ATG” and “ATA”.

Clearly, a graph with vertex orientations can be easily converted to a genome variation graph without vertex orientation by making two copies of each vertex, one for the forward orientation and one for the reverse-complemented orientation. The directed edges with orientations now becomes normal directed edges between corresponding vertex copies. Thus, GFA representation has the same expressive power as the usual directed graphs but with the additional advantage that vertex orientations allow a more compact specification.

2 Highly variant repeat unit details from STR VCF of the 1000 Genomes data

Table 2. Details of the 20 selected Repeat Units (RU)s from the STR VCF. The second column indicates the location of the RU in the chromosome. The third column indicates the range of repeat per allele (RPA) values for the RU as observed in the VCF data. The fourth column gives the variance of all the observed RPA values for the RU in the VCF data.

Chromosome	**Position**	**Repeat Unit**	**Repeat Count Range**	**Variance**
14	26797946	AT	[5, 32]	65.25
13	19091549	AT	[2, 28]	64.32
11	55651634	TA	[1, 27]	61.28
1	231667878	AC	[9, 36]	60.69
10	70136200	TA	[6, 32]	60.67
11	72853857	GT	[10, 36]	60.67
11	93247541	TA	[5, 31]	60.67
12	7818349	AC	[6, 32]	60.67
12	64433409	GT	[7, 33]	60.67
13	73418867	AC	[11, 37]	60.67
15	40846105	GT	[6, 32]	60.67
17	18034132	GT	[7, 33]	60.67
19	51348399	TA	[6, 32]	60.67
2	91796715	TA	[3, 29]	60.67
15	20946820	TG	[6, 32]	58.44
13	95118223	TA	[7, 33]	56.45
14	19609044	TG	[4, 29]	56.25
14	68727016	GT	[7, 32]	56.25
15	77997517	AC	[7, 32]	56.25
10	89558383	GT	[7, 33]	55.28

3 Statistics of the 20 candidate subgraphs generated from the 1000 Genomes data

Table 3. Statistics of candidate 1000 Genomes variation graphs used for evaluating V-ALIGN. The location of the feedback vertex in each graph is given inside bracket in the second column. The last column gives the number of 1000 Genomes VCF entries (variants) captured in the graph.

Graph-ID	#V′ (Vertex-ID)	#VCF entries
14.1	1 (1448662)	266
1	1 (23091037)	281
11.1	1 (8042717)	254
11.2	1 (10311682)	258
12.1	1 (876689)	270
12.2	1 (6931935)	257
13.1	1 (602410)	266
13.2	1 (6560645)	260
13.3	1 (8938196)	280
15.1	1 (704440)	168
15.2	1 (2772289	264
17	1 (2099434)	261
19	1 (5749514)	284
2	1 (26492433)	123
10.1	1 (7624875)	254
10.2	1 (9755781)	297
11.2	1 (6118582)	273
14.3	1 (651187)	228
14.2	1 (6053225)	262
15.3	1 (6843734)	286

The following plot gives the vertex and edge statistics for the 20 candidate graphs. Here, V_a and E_a are the number of vertices and edges in the corresponding graph with single literal vertex labels.

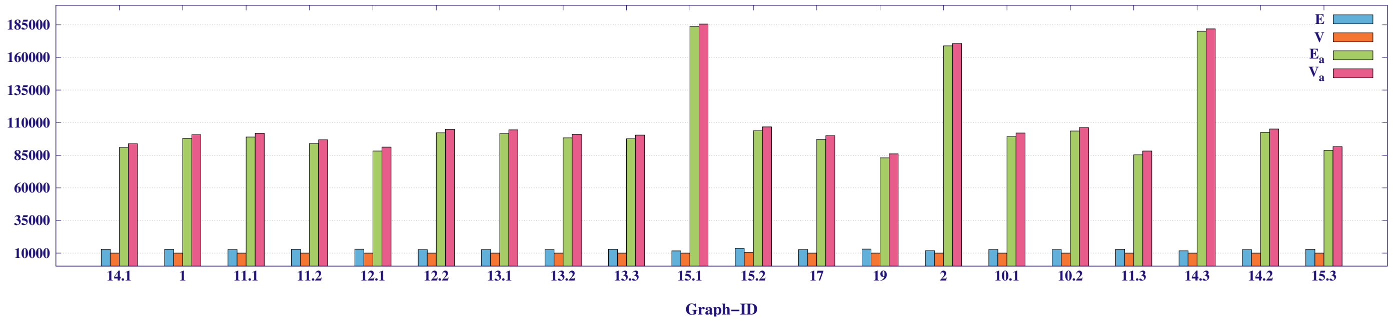

Fig 2. Edge and vertex statistics of the 20 candidate graphs. The Y-axis denotes the count.

4 Generation of seed query sequences for 1000 Genomes variation graphs

Following command is used to generate alternate seed sequence from the 1000 length sequence window.

Command:

```
java -jar GenomeAnalysisTK.jar
     -T FastaAlternateReferenceMaker
     -R reference.fasta
     -L input.intervals
     -V input.vcf
     -o output.fasta
```

Table 4 gives the parameters fed to the command for all the 20 graph instances respectively.

Table 4. Parameters fed to GATK toolkit for seed query sequence generation

Graph-ID	R	L	V	o
14.1	14.fa	chr14:26797446-26798446	14.vcf	14.1.seq
1	1.fa	chr1:231667378-231668378	1.vcf	1.seq
11.1	11.fa	chr11:72853357-72854357	11.vcf	11.1.seq
11.2	11.fa	chr11:93247041-93248041	11.vcf	11.2.seq
12.1	12.fa	chr12:7817849-7818849	12.vcf	12.1.seq
12.2	12.fa	chr12:64432909-64433909	12.vcf	12.2.seq
13.1	13.fa	chr13:19091049-19092049	13.vcf	13.1.seq
13.2	13.fa	chr13:73418367-73419367	13.vcf	13.2.seq
13.3	13.fa	chr13:95117723-95118723	13.vcf	13.3.seq
15.1	15.fa	chr15:20946320-20947320	15.vcf	15.1.seq
15.2	15.fa	chr15:40845605-40846605	15.vcf	15.2.seq
17	17.fa	chr17:18033632-18034632	17.vcf	17.seq
19	19.fa	chr19:51347899-51348899	19.vcf	19.seq
2	2.fa	chr2:91796215-91797215	2.vcf	2.seq
10.1	10.fa	chr10:70135700-70136700	10.vcf	10.1.seq
10.2	10.fa	chr10:89557883-89558883	10.vcf	10.2.seq
11.3	11.fa	chr11:55651134-55652134	11.vcf	11.3.seq
14.3	14.fa	chr14:19608544-19609544	14.vcf	14.3.seq
14.2	14.fa	chr14:68726516-68727516	14.vcf	14.2.seq
15.3	15.fa	chr15:77997017-77998017	15.vcf	15.3.seq

5 V-ALIGN Usage

V-ALIGN provides the following command line options for aligning any length sequence to a genome variation graph.

- **-g filePath:** tag g or G takes input genome variation graph file which can be either an adjacency file or a GFA file or a simple DOT file.
- **-x filePath:** tag x or X takes input sequence file. Input sequence file can have any number of input sequences separated by a new line.
- **-go realNumber:** tag go or GO takes gap open cost. By default the gap open cost is 10.0.
- **-ge realNumber:** tag ge or GE takes gap extension cost. By default the gap extension cost is 0.5.
- **-global:** tag global or GLOBAL provides end-to-end alignment of the input sequences. By default V-ALIGN performs this alignment.

- **-local:** tag local or LOCAL provides local alignment of the input sequences.
- **-o filePath:** tag o or O generates an output file with alignment results. By default the output will be stored in a "out.txt" file in the current directory.
- **-d filePath:** tag d or D generates debug information. By default the output will be stored in a "debug.txt" file in the current directory.
- **-v filePath:** tag v or V takes Feedback Vertex Set (FVS) file.
- **-dot directoryPath:** tag dot or DOT generates DOT files for visualizing the alignment results of input sequences. It will create a folder with a suffix "DotVisuals" in the directory given in the path and stores the DOT files and shell script in the created directory.

6 Alignment visualization

V-ALIGN generates dot files for visualizing the alignment results. The alignment result shows the alignment between the input sequences and the optimal alignment path in the target graph.

Following is the color coding used by V-ALIGN to show the gapped alignment.

- BLACK: Black indicates exact match of the input sequence symbols to symbols on the graph path.
- RED: Red indicates deletion along the graph path.
- *Hyphen* $(-)$: Hyphen represents deletion along the input sequence.
- BLUE: Blue represents a substitution.
- GRAY: Gray represents the prefix and suffix of a sequence that are excluded from the alignment.
- VIOLET: Violet colored vertex number represents the reverse-complemented sequence of the corresponding original vertex.

Figure 3A shows a simple genome variation graph on 2 vertices and one edge. Figure 3B shows the visualization of alignment result generated by V-ALIGN for the input sequence $ATGCATGCATGCAGATCGATCGGGAT$ with the -global option.

Figure 4 shows the visualization of alignment computed by V-ALIGN to a graph that contains cycles. Figure 4A shows a graph on three vertices and having girth 2. Figure 4B shows an optimal alignment path in this graph that was computed by V-ALIGN for

(A) A simple genome variation graph

(B) Visualization of the alignment

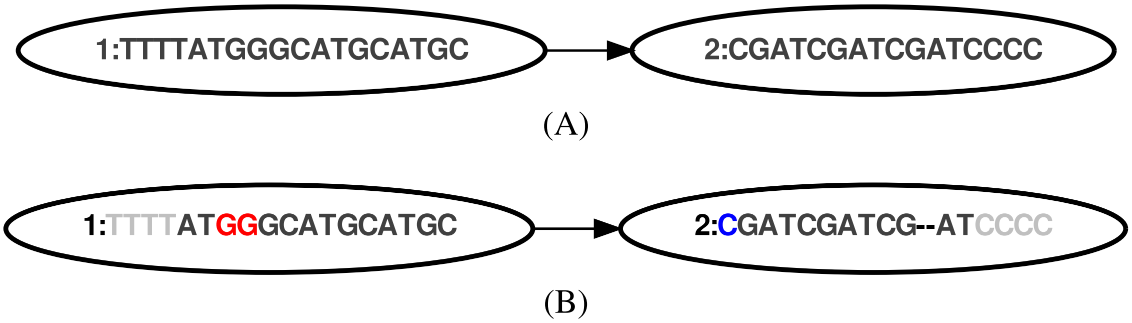

Fig 3. A genome variation graph and the alignment result using V-ALIGN for the input sequence *ATGCATGCATGCAGATCGATCGGGAT* .

aligning the input sequence *GTCGGCGTGA*. The alignment path is color coded using the coloring scheme described earlier. The graph vertices appear multiple times in the shown alignment path because V-ALIGN has traversed through the graph cycle multiple times to identify optimal alignment path.

(A) Sample input graph Input graph of 3 vertices, 4 edges and having girth 2 (cycle involving vertices 2 and 3).

(B) V-ALIGN alignment. Visualization of the alignment of the input sequence *GTCGGCGTGA* on the graph 4A using V-ALIGN. Blue colored vertices indicate substituion.

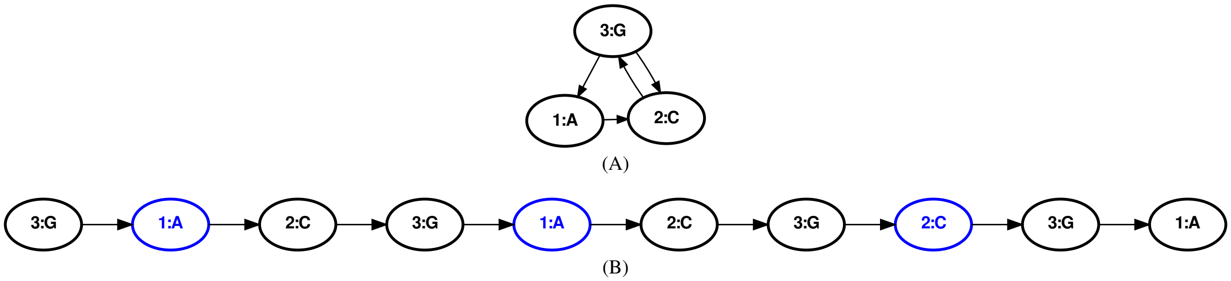

Fig 4. Illustration of sequence alignment using V-ALIGN.

7 DAGification visualization

We show the visualization of the intermediate graphs resulting from the DAGification preprocessing which is performed by POA based alignment approaches. Figures 5, 6 and

7 respectively show the k-DAGified intermediate graphs for $k = 10, 25, 50$ for the input graph in Fig 4A. Figure 5 also shows a color coded optimal alignment path present in the intermediate graph for the input sequence $GTCGGCGTGA$. The many-to-one mapping that exists from the vertices of the DAGified graph to the vertices of the input graph gives the corresponding alignment path in the input graph.

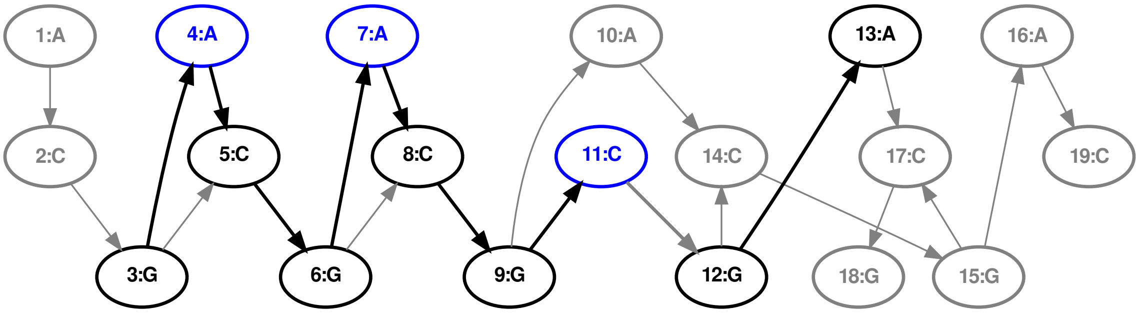

Fig 5. k-DAGified graph for $k = 10$. The input graph is Fig 4A. The DAGified graph contains 19 vertices and 22 edges. The highlighted path from vertex '$3\!:\!G'$ to vertex '$13\!:\!A'$ indicates an optimal alignment path in this graph for the input sequence $GTCGGCGTGA$. The blue colored vertices indicate substitution.

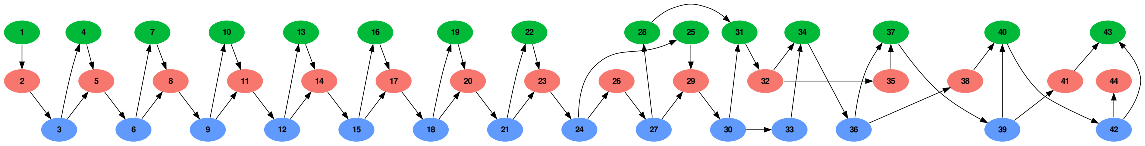

Fig 6. k-DAGified graph for $k = 25$. The graph contains 44 vertices and 56 edges. Vertices with the same color are copies of the same vertex in the input graph.

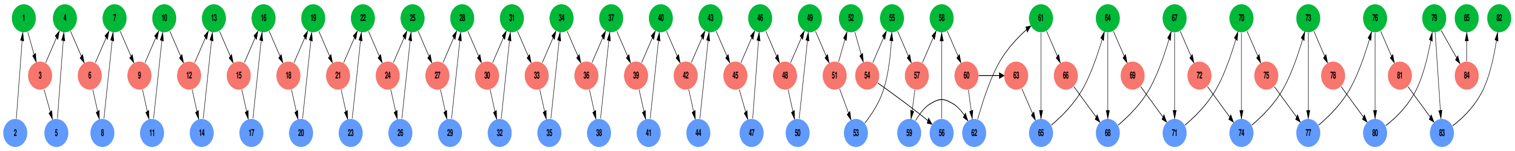

Fig 7. k-DAGified graph for $k = 50$. The graph contains 85 vertices and 110 edges. Vertices with the same color are copies of the same vertex in the input graph.

References

GFA. https://github.com/GFA-spec/GFA-spec, 2017. [Online; accessed 15-April-2017].

Adam M Novak, Glenn Hickey, Erik Garrison et al. Genome graphs. *bioRxiv*, 2017.