Computational Pangenomics #CPANG18

Day 3 (March 8, 2018)

Tobias Marschall, Erik Garrison, and Jordan Eizenga

Wrap up of day 2

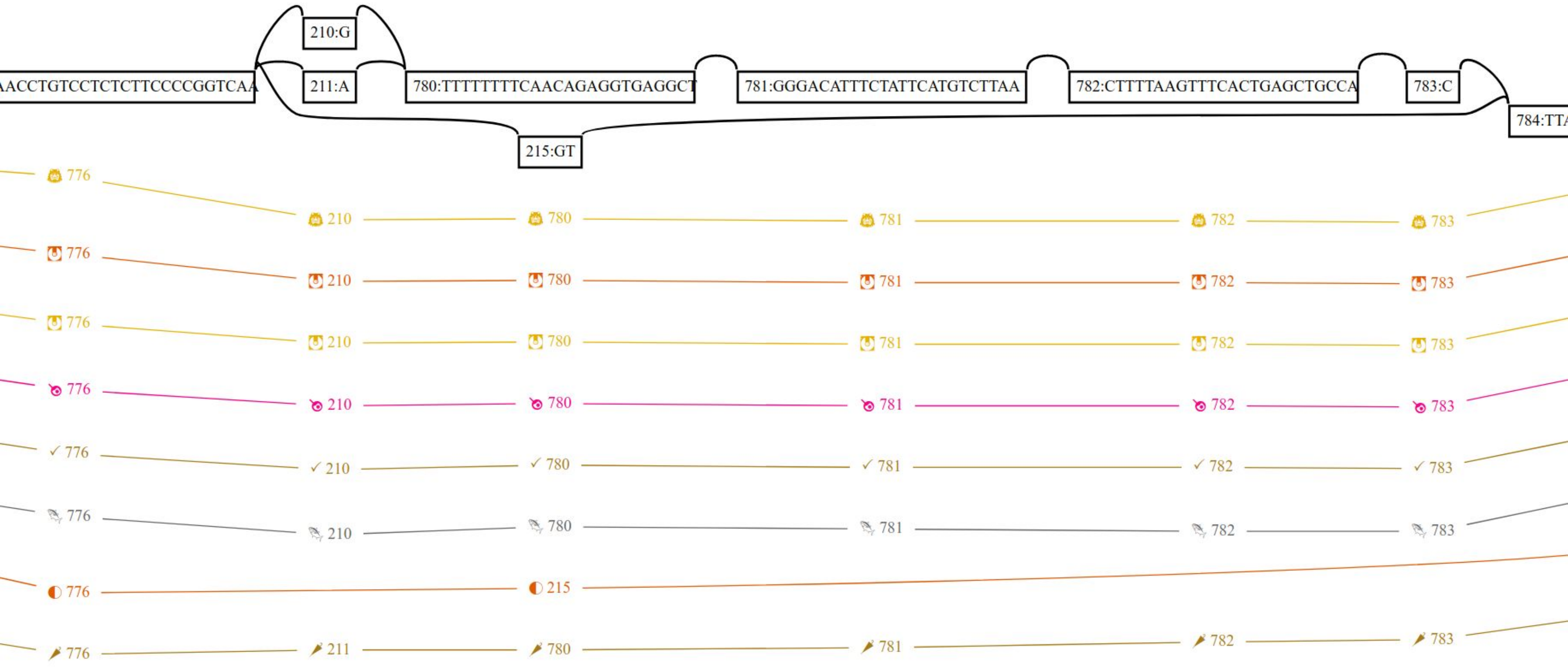
New commands in vg: msga, surject, vectorize, mod, prune, augment

New challenges: building and evaluating pangenome graphs for 5 HIV quasispecies

vg msga

```
vg msga -f GRCh38_alts/FASTA/HLA/L-3139.fa -D \
| vg mod -U 10 - | vg mod -X 32 -c - >L-3139.vg
```

```
vg view -dp L-3139.vg
```



AACCTGTCCTCTCTTCCCCGGTCAA
210:G
211:A
215:GT
780:TTTTTTTTCAACAGAGGTGAGGCT
781:GGGACATTTCTATTCATGTCTTAA
782:CTTTTAAGTTTCACTGAGCTGCCA
783:C
784:TTA
776
210
211
215
780
781
782
783

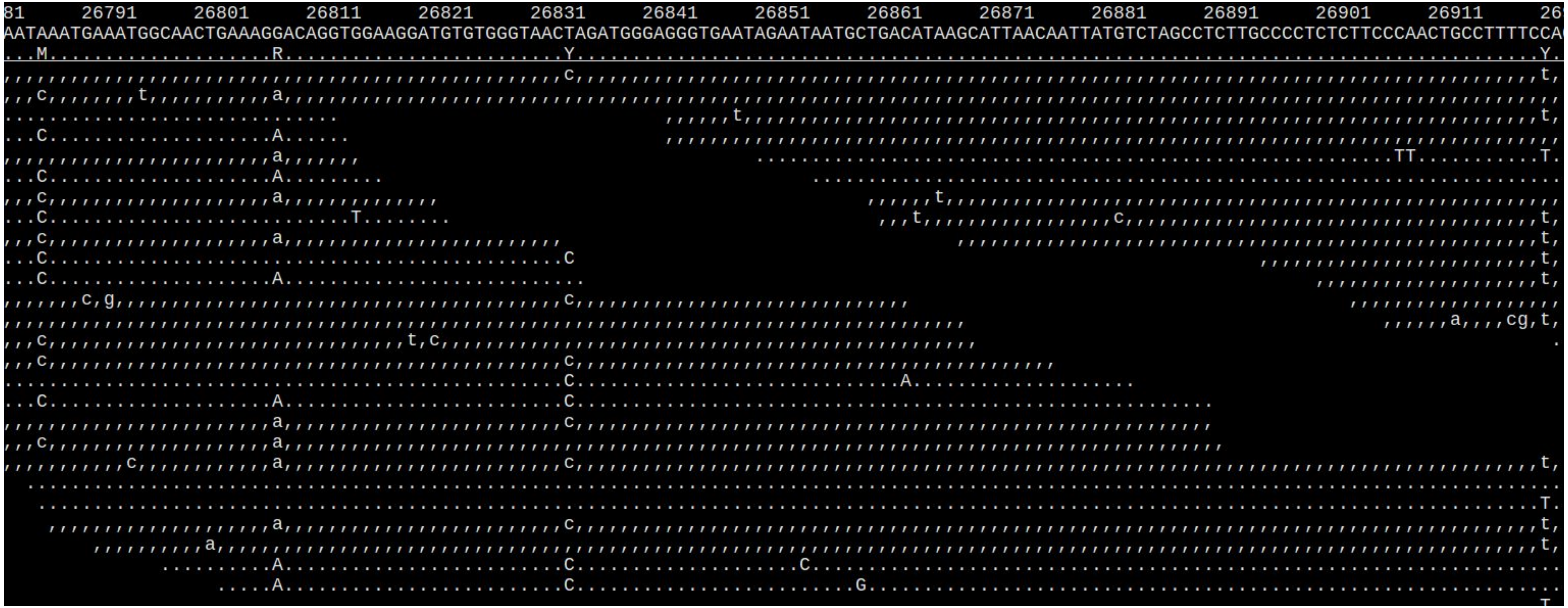
vg surject

vg map -x z.xg -g z.gcsa -G z.sim | vg surject -d z.xg >aln.bam

or

vg map -d z -G z.sim --surject-to bam >aln.bam

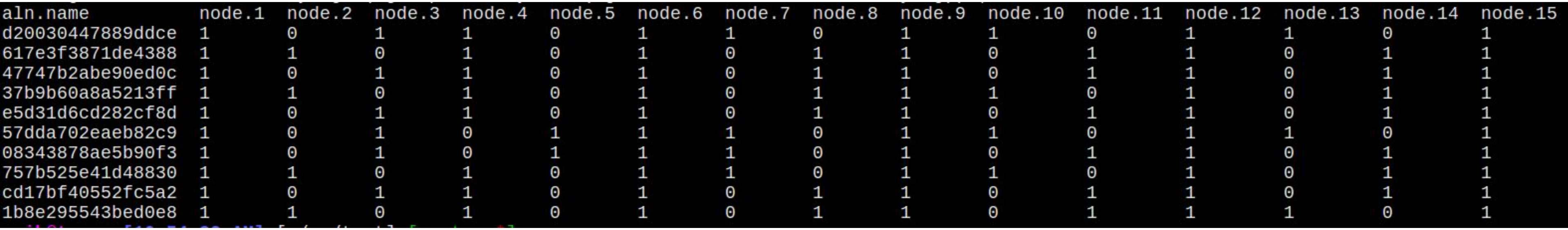
Select the path to surject into with --into-path or --into-paths



```
81      26791     26801     26811     26821     26831     26841     26851     26861     26871     26881     26891     26901     26911     26
AATAAATGAAATGGCAACTGAAAGGACAGGTGGAAGGATGTGTGGGTAACTAGATGGGAGGGTGAATAGAATAATGCTGACATAAGCATTAACAATTATGTCTAGCCTCTTGCCCCTCTCTTCCCAACTGCCTTTTCCAC
...M....................R.........................Y..........................................................................................Y
```

vg vectorize

vg vectorize -f -x tiny.xg aln.gam



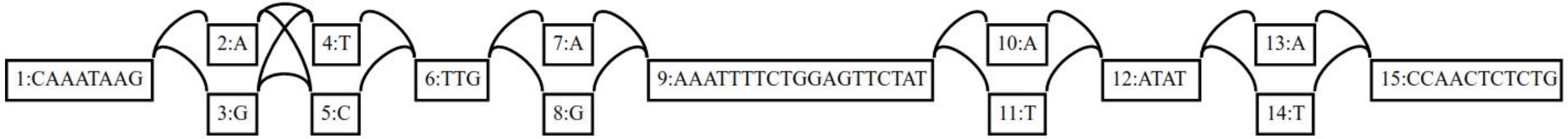
```
aln.name          node.1  node.2  node.3  node.4  node.5  node.6  node.7  node.8  node.9  node.10  node.11  node.12  node.13  node.14  node.15
d20030447889ddce  1       0       1       1       0       1       1       0       1       1        0        1        1        0        1
617e3f3871de4388  1       1       0       1       0       1       0       1       1       0        1        1        0        1        1
47747b2abe90ed0c  1       0       1       1       0       1       0       1       1       0        1        1        0        1        1
37b9b60a8a5213ff  1       1       0       1       0       1       0       1       1       1        0        1        0        1        1
e5d31d6cd282cf8d  1       0       1       1       0       1       0       1       1       0        1        1        0        1        1
57dda702eaeb82c9  1       0       1       0       1       1       1       0       1       1        0        1        1        0        1
08343878ae5b90f3  1       0       1       0       1       1       1       0       1       0        1        1        0        1        1
757b525e41d48830  1       1       0       1       0       1       1       0       1       1        0        1        0        1        1
cd17bf40552fc5a2  1       0       1       1       0       1       0       1       1       0        1        1        0        1        1
1b8e295543bed0e8  1       1       0       1       0       1       0       1       1       0        1        1        1        0        1
```

vg mod

MANY graph modification tools in one command line utility.

- Sorting
- Chopping
- Simplification
- Augmentation
- Unfolding/unrolling
- Path manipulation (add, remove, keep)
- … etc, etc

vg mod -pl / vg prune

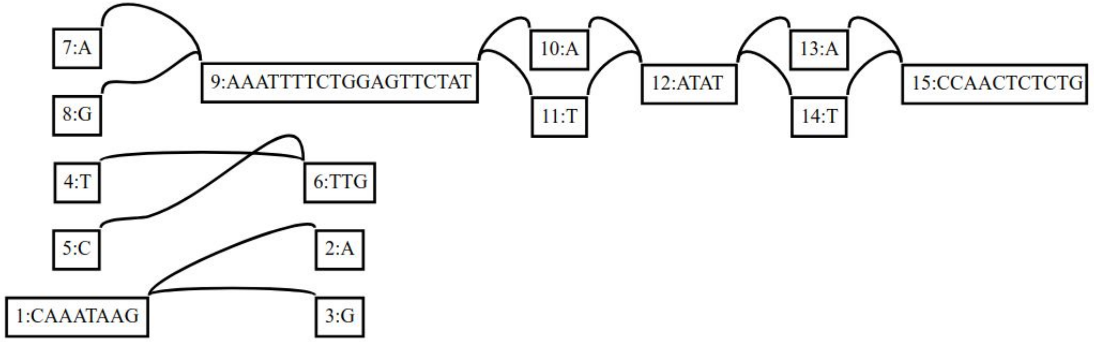


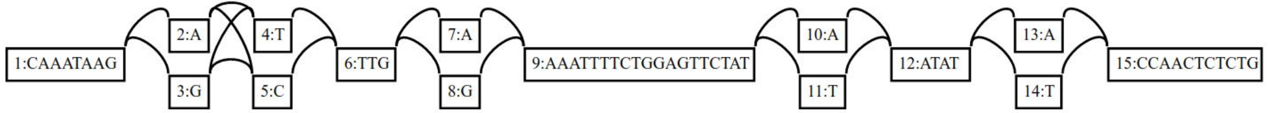
vg mod -pl 8 -e 2 tiny.vg

or

vg prune -k 8 -e 2 -s 0 tiny.vg

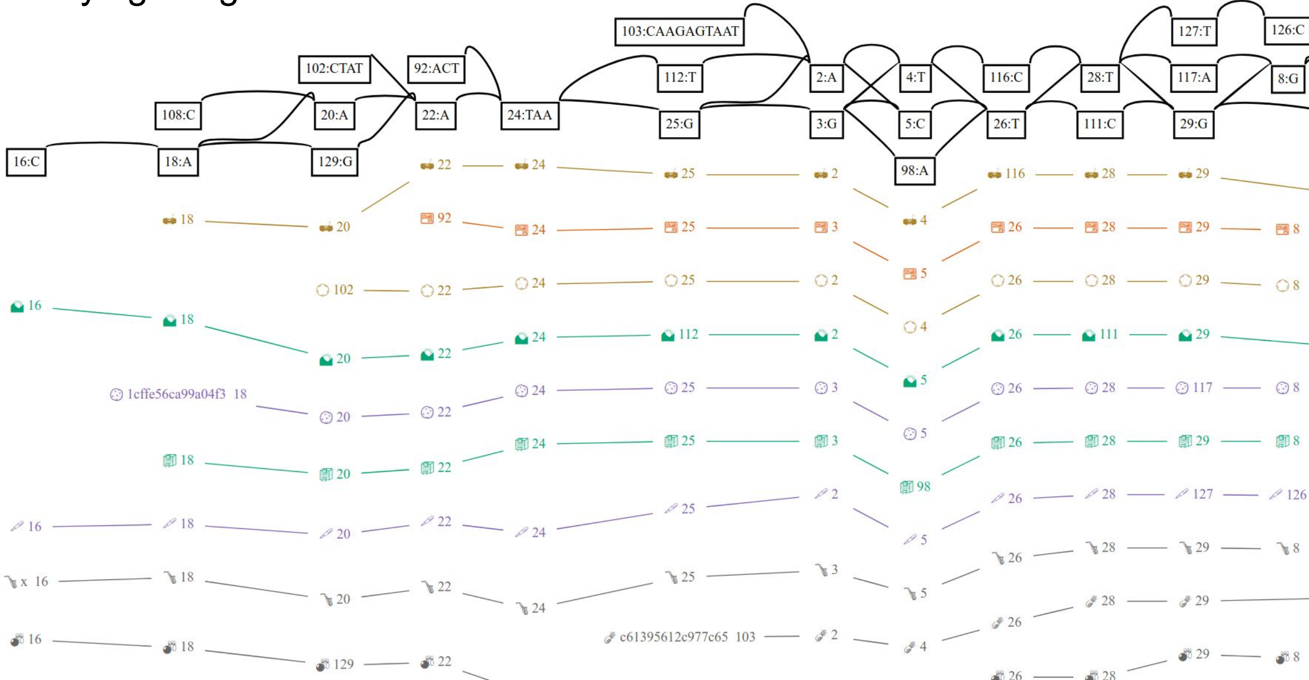
Removes edges for which we would have crossed 2 bifurcations in a path of 8 bases. (Used in indexing.)

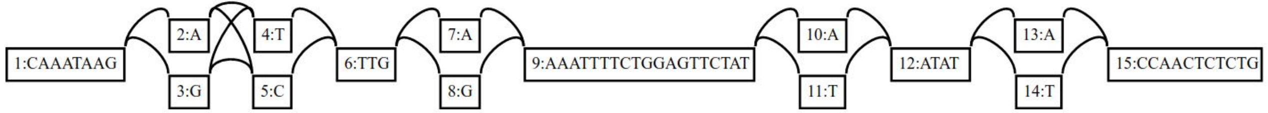




vg mod -i / vg augment

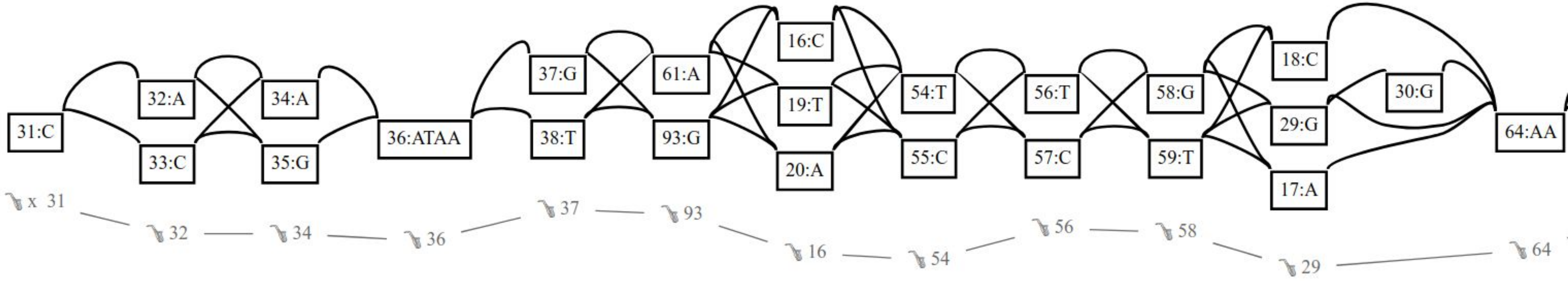
vg map -d tiny -G <(vg sim -n 10 -e 0.1 -i 0.05 -l 50 -a -x tiny.xg) >aln.gam
vg mod -i aln.gam tiny.vg >tiny+.vg
vg augment -g 1 -A aln+aug.gam tiny.vg aln.gam





vg mod -i / vg augment

```
vg map -d tiny -G <(vg sim -n 10 -e 0.1 -i 0.05 -l 50 -a -x tiny.xg) >aln.gam
vg mod -i aln.gam tiny.vg >tiny+.vg
```
vg augment -g 1 -A aln+aug.gam tiny.vg aln.gam



Questions

How confident are you with building your own workflows within vg framework?

How confident are you creating graphs using vg msga?

How confident are you with modifying graphs using vg mod?

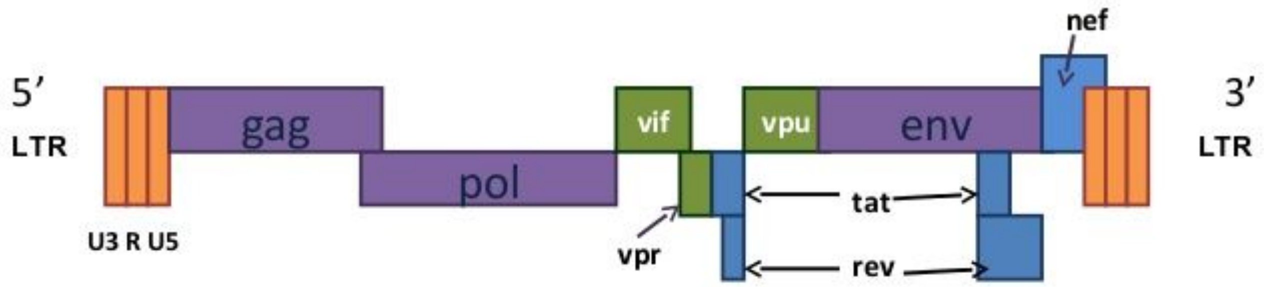
How confident are you with exploring the properties of mixed sequencing data sets using genome graphs?

How well do you understand algorithmic challenges that come with transitioning from linear reference genomes to graph genomes?

Presentations on HIV experiments

Automatic circularization of vg msga progressive assembly resulted from HIV’s ~700bp LTR

HIV Genome



gag – core proteins (including p24)

pol – envelope glycoproteins

env – enzymes (RT, protease, Integrase)

tat, rev, nef, vif, vpu, vpr – proteins in the modification of host cell to enhance virus growth and regulate viral gene expression

LTR – Long Terminal Repeats- for initiation of transcription.

Maybe we’re at coffee break?

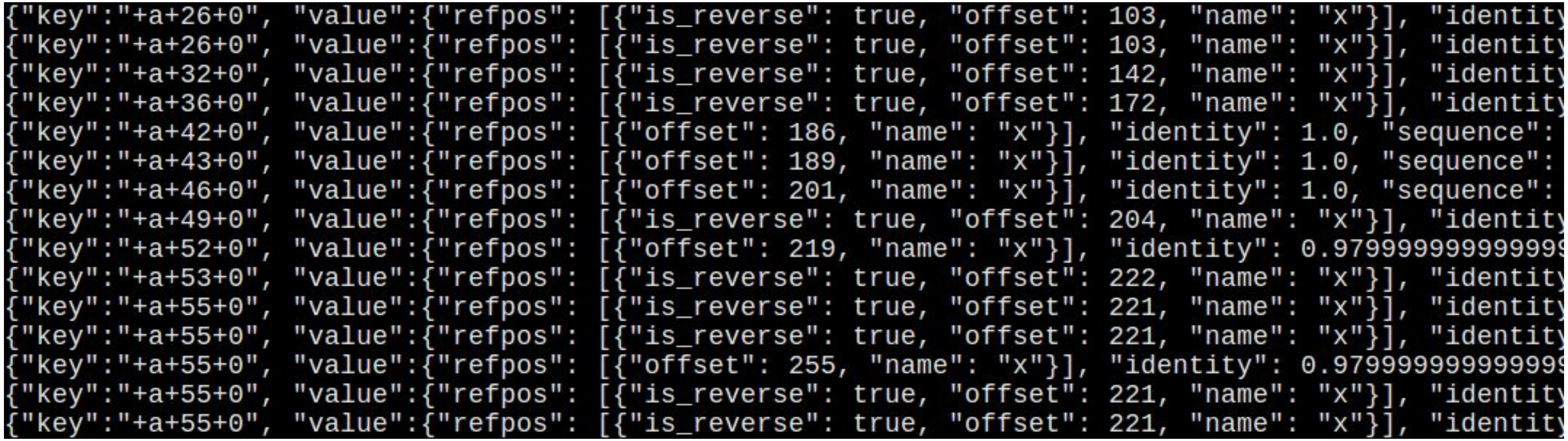
Classical (bacterial) pangenomics

New commands

index (of alignments), chunk, call, genotype, pack

vg index -a (start-node sorted alignment index)

```
vg construct -r small/x.fa -v small/x.vcf.gz >x.vg
vg index -x x.xg -g x.gcsa -k 16 x.vg
vg map -d x -G <(vg sim -n 100 -e 0.01 -i 0.005 -l 50 -a -x x.xg) >aln.gam
```
`vg index -d aln.gam.idx -a aln.gam`
```
vg index -d aln.gam.idx -D
```



```
{"key":"+a+26+0", "value":{"refpos": [{"is_reverse": true, "offset": 103, "name": "x"}], "identity
{"key":"+a+26+0", "value":{"refpos": [{"is_reverse": true, "offset": 103, "name": "x"}], "identity
{"key":"+a+32+0", "value":{"refpos": [{"is_reverse": true, "offset": 142, "name": "x"}], "identity
{"key":"+a+36+0", "value":{"refpos": [{"is_reverse": true, "offset": 172, "name": "x"}], "identity
{"key":"+a+42+0", "value":{"refpos": [{"offset": 186, "name": "x"}], "identity": 1.0, "sequence":
{"key":"+a+43+0", "value":{"refpos": [{"offset": 189, "name": "x"}], "identity": 1.0, "sequence":
{"key":"+a+46+0", "value":{"refpos": [{"offset": 201, "name": "x"}], "identity": 1.0, "sequence":
{"key":"+a+49+0", "value":{"refpos": [{"is_reverse": true, "offset": 204, "name": "x"}], "identity
{"key":"+a+52+0", "value":{"refpos": [{"offset": 219, "name": "x"}], "identity": 0.97999999999999
{"key":"+a+53+0", "value":{"refpos": [{"is_reverse": true, "offset": 222, "name": "x"}], "identity
{"key":"+a+55+0", "value":{"refpos": [{"is_reverse": true, "offset": 221, "name": "x"}], "identity
{"key":"+a+55+0", "value":{"refpos": [{"is_reverse": true, "offset": 221, "name": "x"}], "identity
{"key":"+a+55+0", "value":{"refpos": [{"offset": 255, "name": "x"}], "identity": 0.97999999999999
{"key":"+a+55+0", "value":{"refpos": [{"is_reverse": true, "offset": 221, "name": "x"}], "identity
{"key":"+a+55+0", "value":{"refpos": [{"is_reverse": true, "offset": 221, "name": "x"}], "identity
```

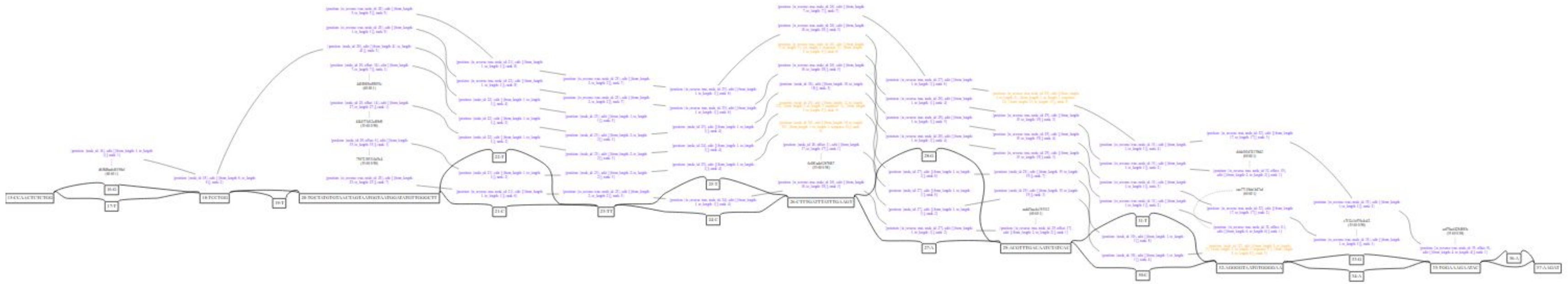
Sorting alignments (by start node id)

```
vg index -A -d aln.gam.idx | vg view -a -
vg index -A -d aln.gam.idx >aln.sort.gam
vg view -a aln.sort.gam | jq '.path.mapping[0].position.node_id' | head
```

```
16
18
20
20
20
22
16
16
32
32
….
```

vg index -N (node to alignment index)

```
vg index -N -d aln.sort.gam.idx aln.sort.gam
vg find -d aln.sort.gam.idx -o 24 | vg view -a - | wc -l
vg find -d aln.sort.gam.idx -o 23 | vg view -a - | wc -l
vg find -x x.xg -n 24 -c 1 >m.vg
vg view -dA  <(vg find -d aln.sort.gam.idx -A m.vg) <(vg find -x x.xg -G
<(vg find -d aln.sort.gam.idx -A m.vg))
```



vg explode (break graphs apart)

```
vg mod -pl 16 -e 3 x.vg | vg explode - parts
parts/component0.vg     x
parts/component1.vg     x
parts/component2.vg     x
parts/component3.vg     x
parts/component4.vg     x
parts/component5.vg
parts/component6.vg     x
parts/component7.vg     x
parts/component8.vg     x
parts/component9.vg     x
```

vg chunk (break graphs into pieces)

```
vg chunk -x x.xg -n 10
ls chunk*
```

```
    chunk_0_ids_1_23.vg
    chunk_1_ids_21_46.vg
    chunk_3_ids_66_90.vg
    chunk_5_ids_109_133.vg
    chunk_7_ids_153_177.vg
    chunk_9_ids_197_210.vg
    chunk_0_ids_1_5_trace_annotate.txt
    chunk_2_ids_43_68.vg
    chunk_4_ids_88_112.vg
    chunk_6_ids_131_155.vg
    chunk_8_ids_175_200.vg
```

vg pack (graph coverage vectors)

```
vg pack -x x.xg -g aln.gam -d -n
```

```
0      0
1      0
2      1
3      0
4      1
5      1
6      2
7      2
8      0
9      2
10      2
11      0
12      2
….
```