VariantStore: A Large-Scale Genomic Variant Search Index

Prashant Pandey
Berkeley Lab

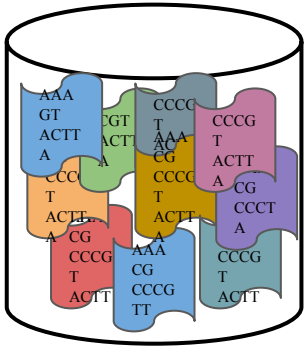
Yinjie Gao
Carnegie Mellon University

Carl Kingsford
Carnegie Mellon University

Country-scale sequencing efforts produce huge amount of gene variation data

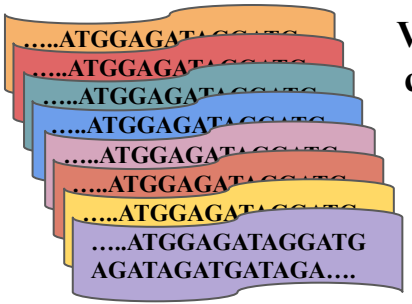
Individuals

Sequencing



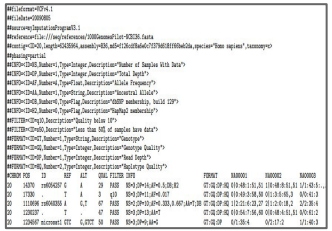
Raw sequencing data

Assembly



Assembled data

Variant calling

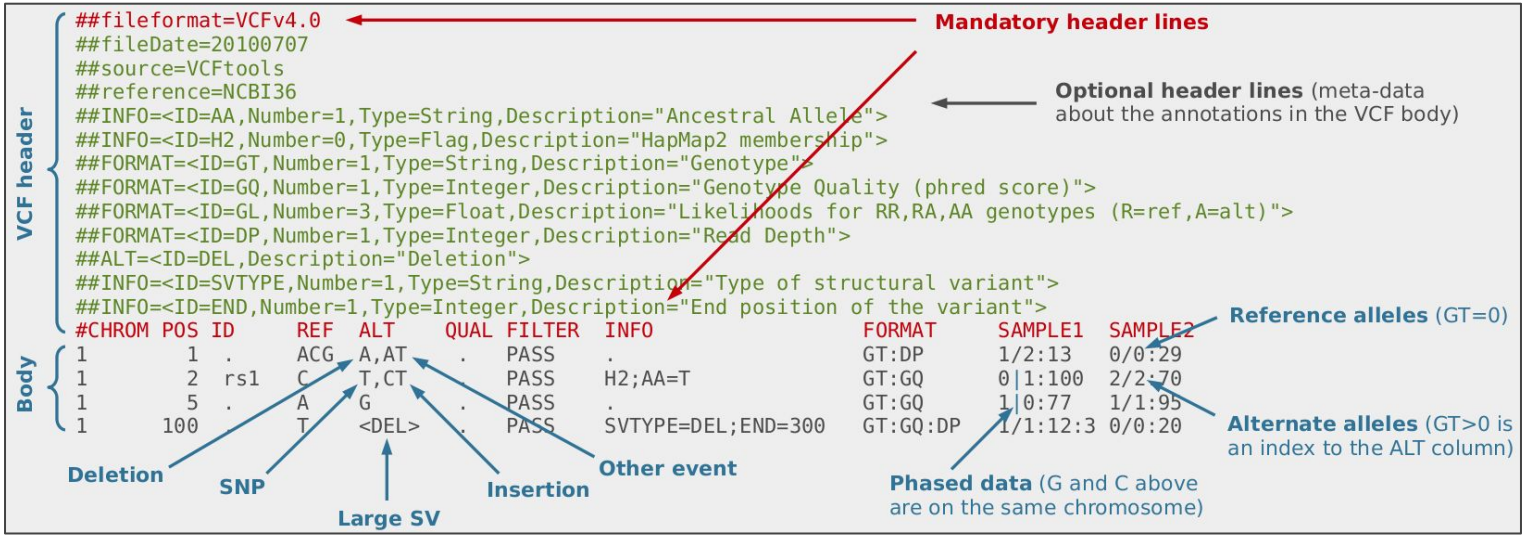


Genomic variants

Variation information can help improve applications

- Tens of millions of genomes are now commonly available
- Genomes contain huge amounts of diversity information
- Variation information promises to improve applications:
  - **Genomes assembly**
  - **Population level disease analysis**
  - **Genome wide association studies (GWAS)**
  - **Personalized medicine**
  - **Predicting remission rate for cancer**

Variant call format (VCF) enodes variation information



- The VCF format has been developed to encode variants from large scale sequencing
- These files contain variations as mutations based on a reference genome
  - SNPs and Indels

Picture taken from: http://vcftools.sourceforge.net/VCF-poster.pdf

Multiple coordinate systems in variation data

Reference sequence		CAATTTGCTGATCT				
Position	Reference seq.	Alternative seq.	HG00096	HG00101	HG00103	Variant type
2	A	G	0	1	1	SUBSTITUTION
2	AATT	A	1	0	0	DELETION
6	T	TACG	0	0	1	INSERTION

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Reference	C	A	A	T	T	T	G	C	T	G	A	T	C	T			
HG00096	C	A	T	G	C	T	G	A	T	C	T						
HG00101	C	G	A	T	T	T	G	C	T	G	A	T	C	T			
HG00103	C	G	A	T	T	T	A	C	G	G	C	T	G	A	T	C	T

- A coordinate system uniquely identifies the position of a variant in a given genome
- Each sample can have a different coordinate system

Multiple coordinate systems in variation data

Reference sequence			CAATTTGCTGATCT			
Position	Reference seq.	Alternative seq.	HG00096	HG00101	HG00103	Variant type
2	A	G	0	1	1	SUBSTITUTION
2	AATT	A	1	0	0	DELETION
6	T	TACG	0	0	1	INSERTION

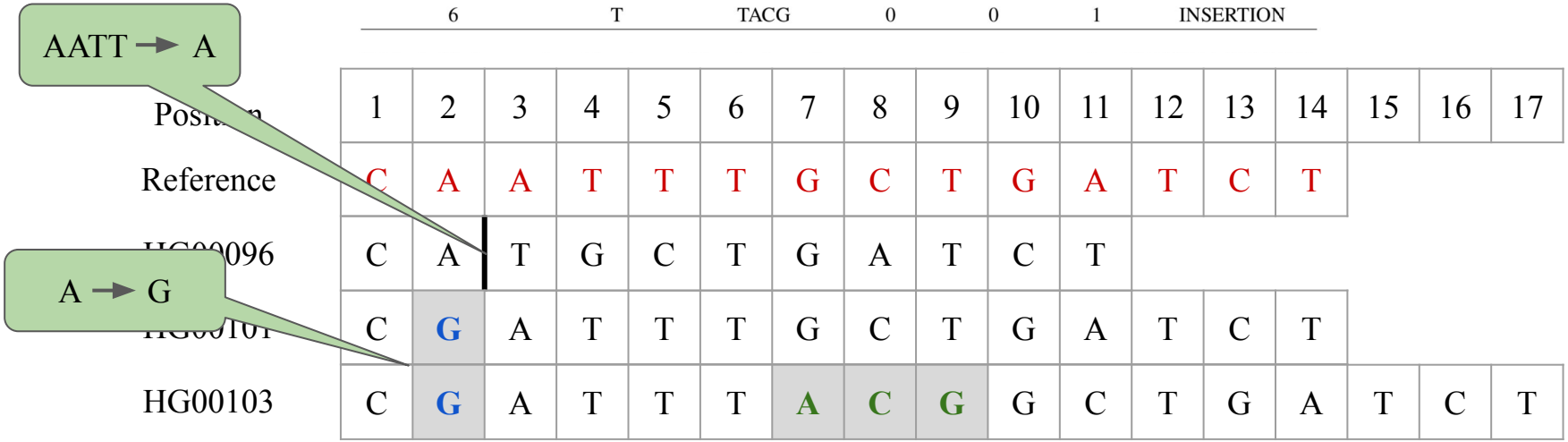
AATT → A

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Reference	C	A	A	T	T	T	G	C	T	G	A	T	C	T			
HG00096	C	A	T	G	C	T	G	A	T	C	T						
HG00101	C	G	A	T	T	T	G	C	T	G	A	T	C	T			
HG00103	C	G	A	T	T	T	A	C	G	G	C	T	G	A	T	C	T

- A coordinate system uniquely identifies the position of a variant in a given genome
- Each sample can have a different coordinate system

Multiple coordinate systems in variation data

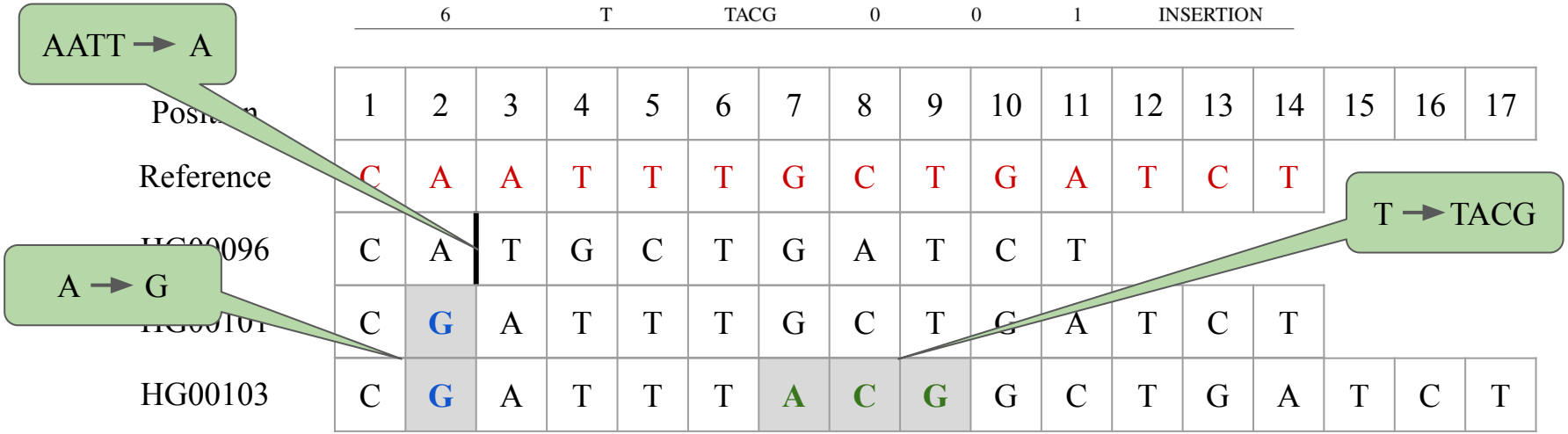
Reference sequence		CAATTTGCTGATCT				
Position	Reference seq.	Alternative seq.	HG00096	HG00101	HG00103	Variant type
2	A	G	0	1	1	SUBSTITUTION
2	AATT	A	1	0	0	DELETION
6	T	TACG	0	0	1	INSERTION



- A coordinate system uniquely identifies the position of a variant in a given genome
- Each sample can have a different coordinate system

Multiple coordinate systems in variation data

Reference sequence		CAATTTGCTGATCT				
Position	Reference seq.	Alternative seq.	HG00096	HG00101	HG00103	Variant type
2	A	G	0	1	1	SUBSTITUTION
2	AATT	A	1	0	0	DELETION
6	T	TACG	0	0	1	INSERTION



- A coordinate system uniquely identifies the position of a variant in a given genome
- Each sample can have a different coordinate system

Different types of variant queries

1. Find the closest variant to position X for all samples in the reference coordinates.
2. Find the sequence between positions X and Y for sample S in the reference coordinates.
3. Find the sequence between positions X and Y for sample S in the sample coordinates.
4. Find all variants between positions X and Y for sample S in the reference coordinates.
5. Find all variants between positions X and Y for sample S in the sample coordinates.
6. Find all variants between positions X and Y for all samples in the reference coordinates.

Need a system to index and query in multiple coordinate systems

- Each sample coordinate system requires a function

$$f(\text{pos}_{REF}) \rightarrow \text{pos}_{SAMPLE}$$

- Maintaining thousands of mappings increases memory footprint and computational complexity
- Limits the scalability of variant indexes to population level data

VariantStore: a system to efficiently index and query population-level variation data

- Supports querying variants in both reference and sample-specific coordinates
  - Takes between **0.002 -- 3 seconds for different types of variant queries**
- Scales to data containing thousands of samples and millions of variants
  - 1000 Genomes project, **2500 samples and 924M variants, 3 Hrs**
  - TCGA (BRCA) project, **8640 samples and 5M variants, 4 Hrs**
- Efficiently scales out-of-RAM to enable memory-efficient construction and query
  - **Peak RAM is 10% the size of the index**

Existing solutions do not scale to thousands of samples

- Existing solutions are built to cater to specific applications
- For example, VG toolkit[1] and Seven Bridges[2] are built for read mapping applications
- They encode variants in a **variation graph** and perform graph traversals for read mapping
- They support sequence search but **do not support other kinds of queries**
- The solutions are **not designed to scale with increasing amounts of population-level variation data**

[1] Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature Biotechnology*, 36:875–879, 2018
[2] Fast and accurate genomic analyses using genome graphs. *Nature Genetics*, 51:354–362, 2019

Reference-only indexes do not support multiple coordinate queries

- GQT[1], BGT[2], and GTC[3] are **reference-only indexes**
- They are optimized to support positional variant queries but **do not store sequences for comparison**
- Traditional database-based solutions have proven **prohibitively slow**

[1] Efficient genotype compression and analysis of large genetic-variation data sets. *Nature Methods*, 13(1):63, 2016
[2] BGT: efficient and flexible genotype query across many samples. *Bioinformatics*, 32(4): 590–592, 2015
[3] GTC: how to maintain huge genotype collections in a compressed form. *Bioinformatics*, 34(11):1834–1840, 2018

VariantStore components

- **Variants are encoded in variation graph**

Variation graphs encode variations based on a reference

- **Variation graph is a directed, acyclic graph (DAG) that embeds a set of genomic sequences**
- Graph $G(N, E, P)$ contains as set of nodes N, set of edges E, and set of paths P
- Each node represents a sequence
- Edges connect nodes containing sequences that are present in genomes
- A path is a set of nodes through the graph that represents a complete genomic sequence

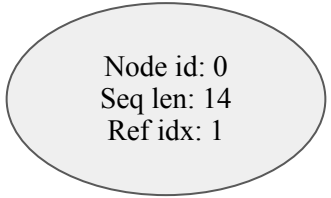
Reference sequence and sample variants

Reference sequence	CAATTTGCTGATCT					
Position	Reference seq.	Alternative seq.	HG00096	HG00101	HG00103	Variant type
2	A	G	0	1	1	SUBSTITUTION
2	AATT	A	1	0	0	DELETION
6	T	TACG	0	0	1	INSERTION

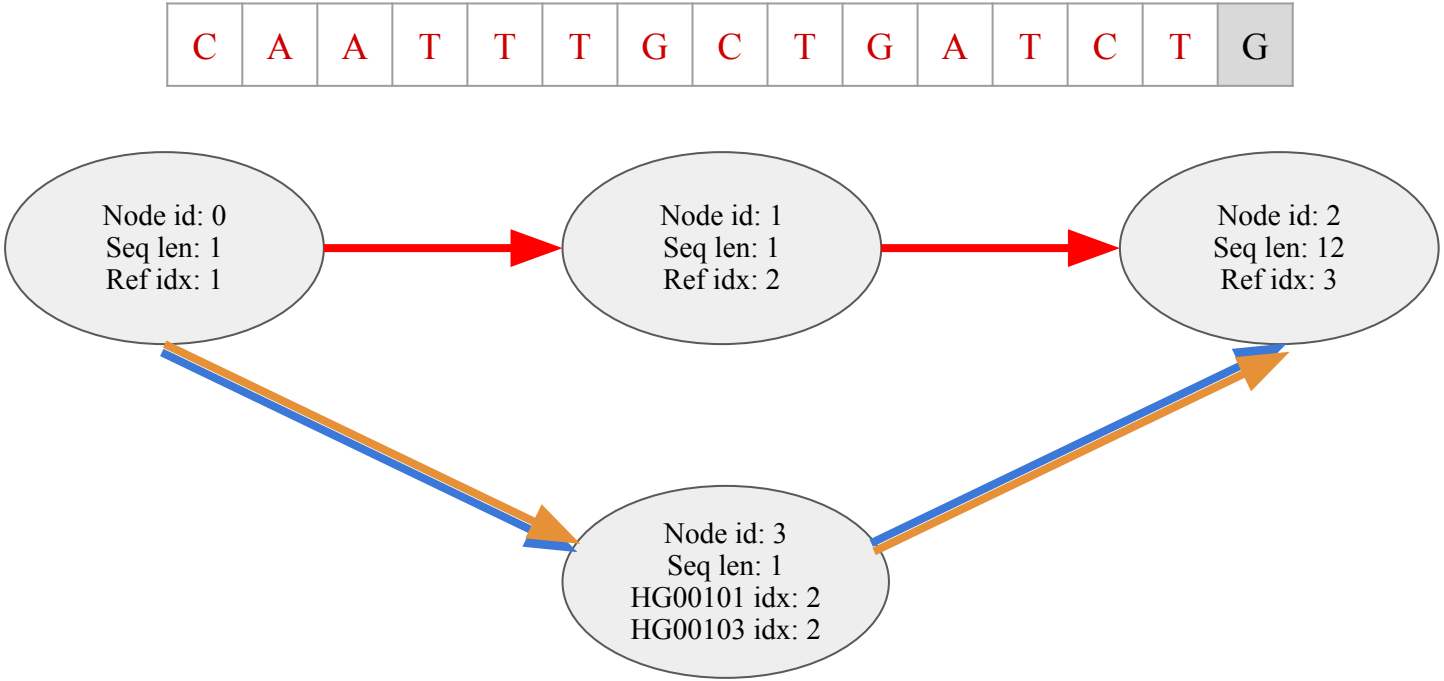
Table 2: Variants ordered by the position in the reference genome for three samples (HG00096, HG00101, HG00103). Each variant has the list of samples that contain the variant.

Variation graph with on the reference sequence

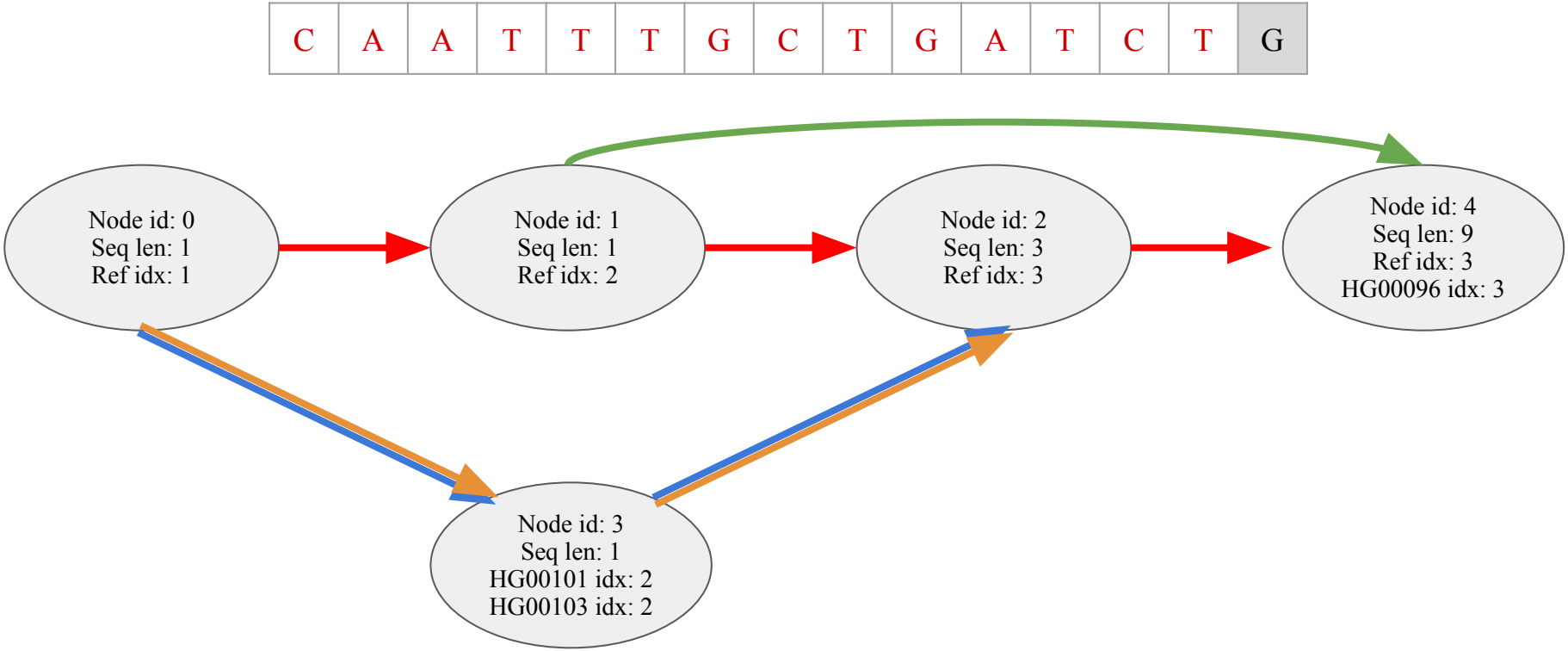
C	A	A	T	T	T	G	C	T	G	A	T	C	T



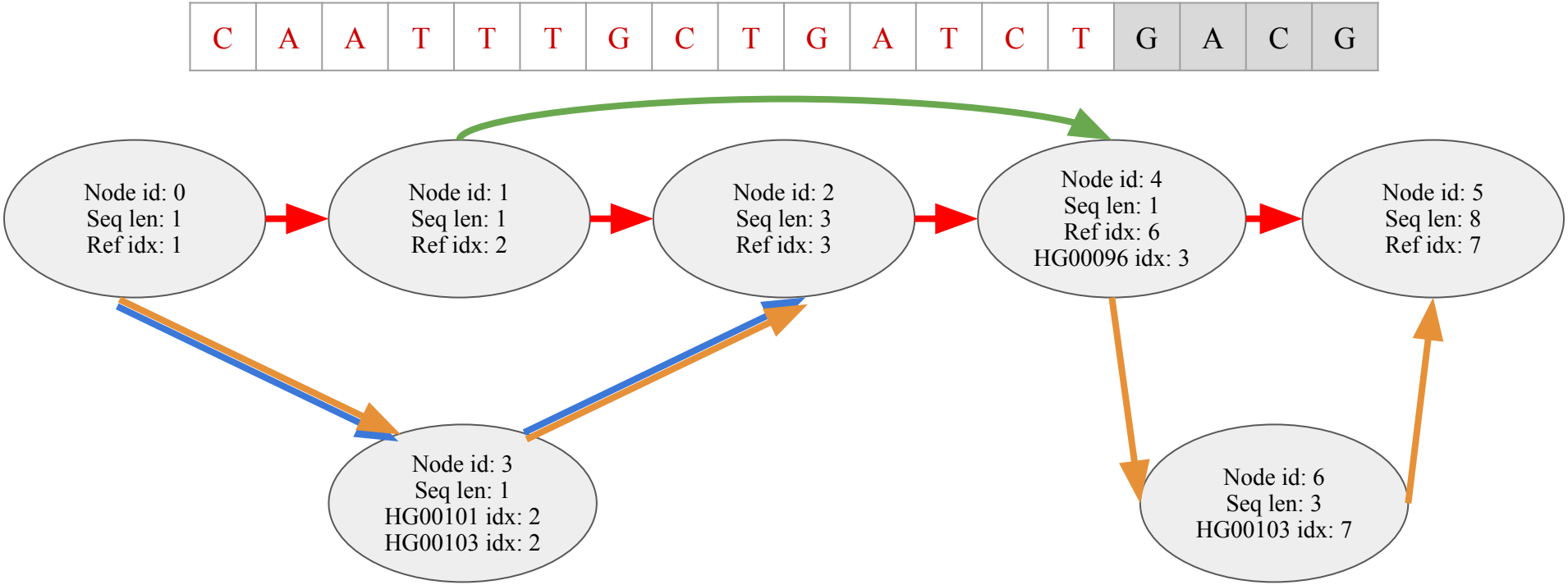
Adding a “substitution” variant in variation graph



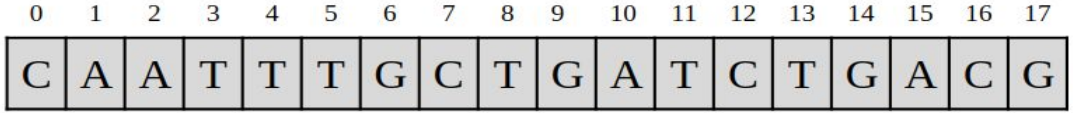
Adding a “deletion” variant in variation graph

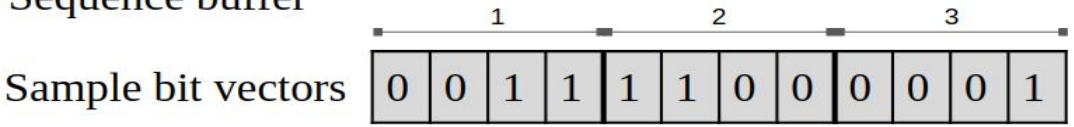


Adding an “addition” variant in variation graph



Variation graph representation in VariantStore

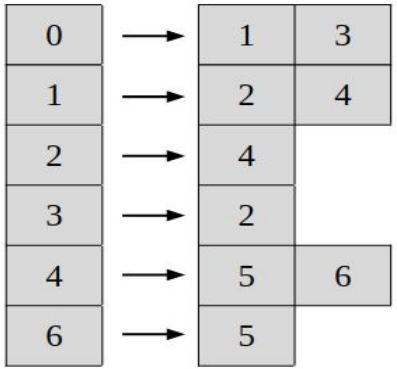




Node list

ID	Offset	Len	Index	Class Id
0	0	1	1	0
1	1	1	2	0
2	2	3	3	0
3	14	1	2, 2	1
4	5	1	6, 3	2
5	6	8	7	0
6	15	3	7	3

Graph topology



Variation graph representation in VariantStore

Same variant is often present in multiple samples

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
C	A	A	T	T	T	G	C	T	G	A	T	C	T	G	A	C	G

Sequence buffer

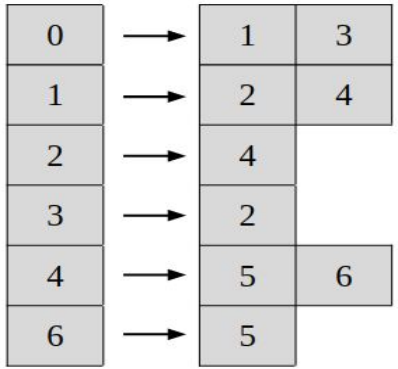
1		2		3	
0	0	1	1	1	1	0	0	0	0	0	1

Sample bit vectors

Node list

ID	Offset	Len	Index	Class Id
0	0	1	1	0
1	1	1	2	0
2	2	3	3	0
3	14	1	2, 2	1
4	5	1	6, 3	2
5	6	8	7	0
6	15	3	7	3

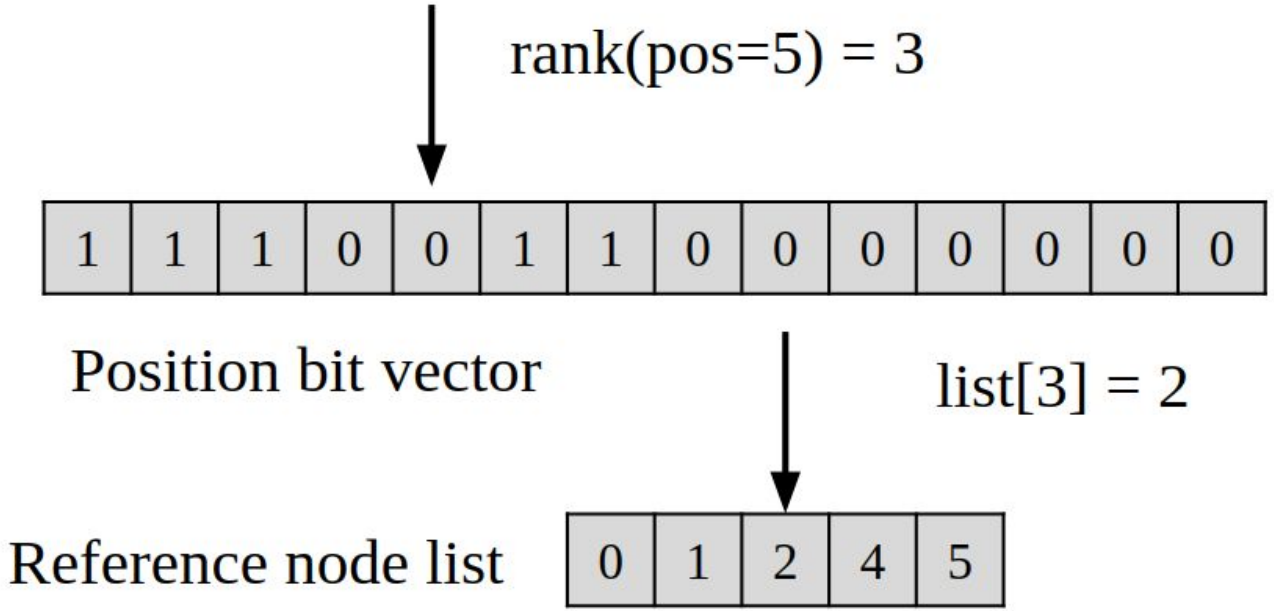
Graph topology



VariantStore components

- Variants are encoded in variation graph
- **Position-based index to locate node in the graph corresponding to a position**

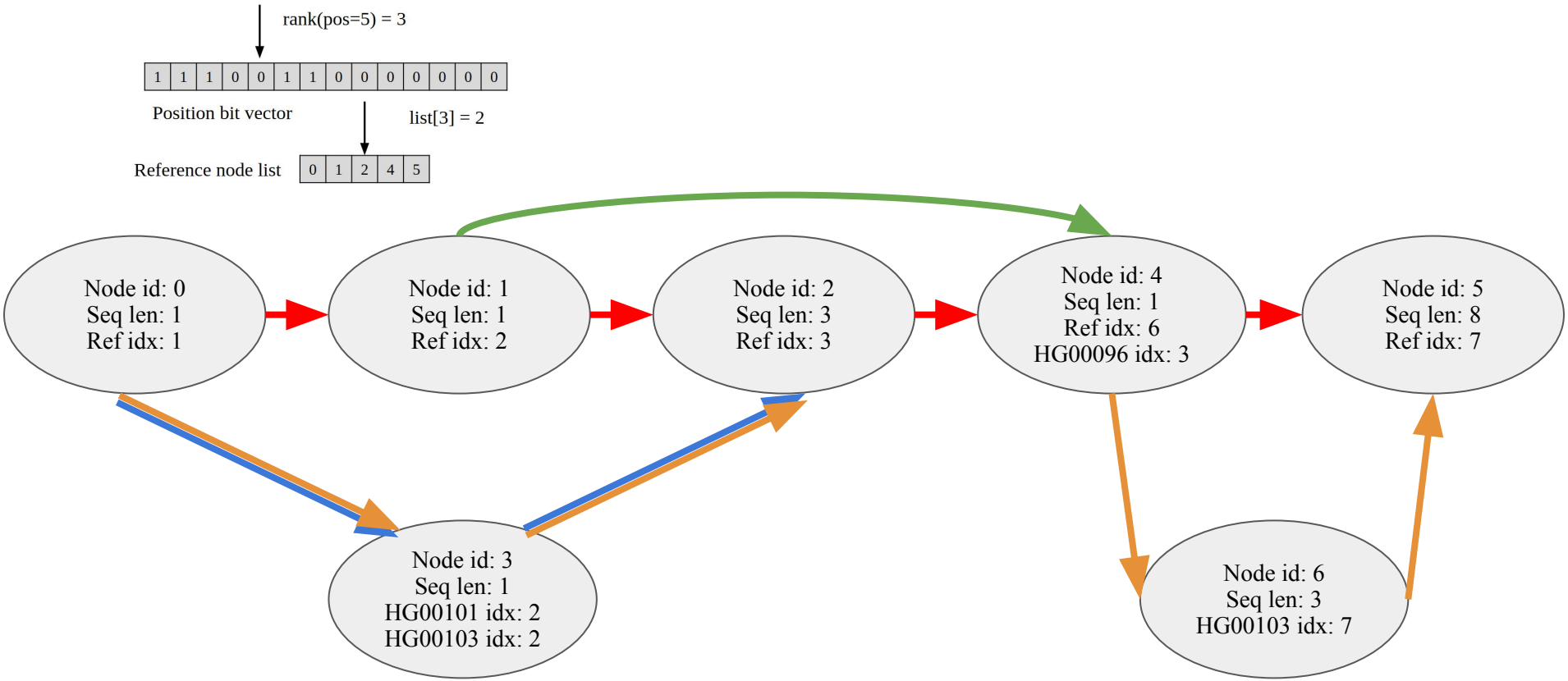
Position index to lookup nodes in variation graph



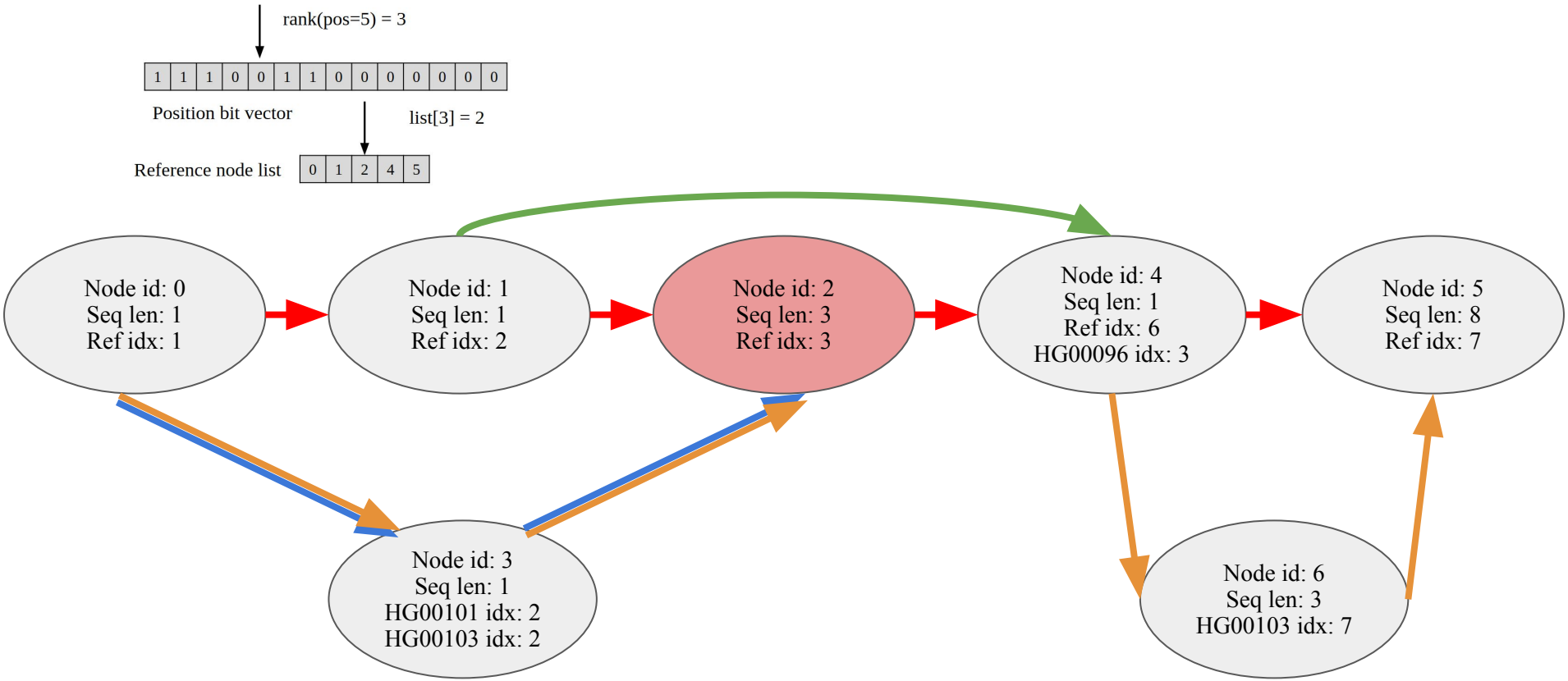
VariantStore components

- Variants are encoded in variation graph
- Position-based index to locate node in the graph corresponding to a position
- **Position index is only maintained for a single reference genome**
- **Use local graph traversals to translate between coordinate systems**
- **Marker nodes in the graph store absolute sample positions and bound local traversals during translation**

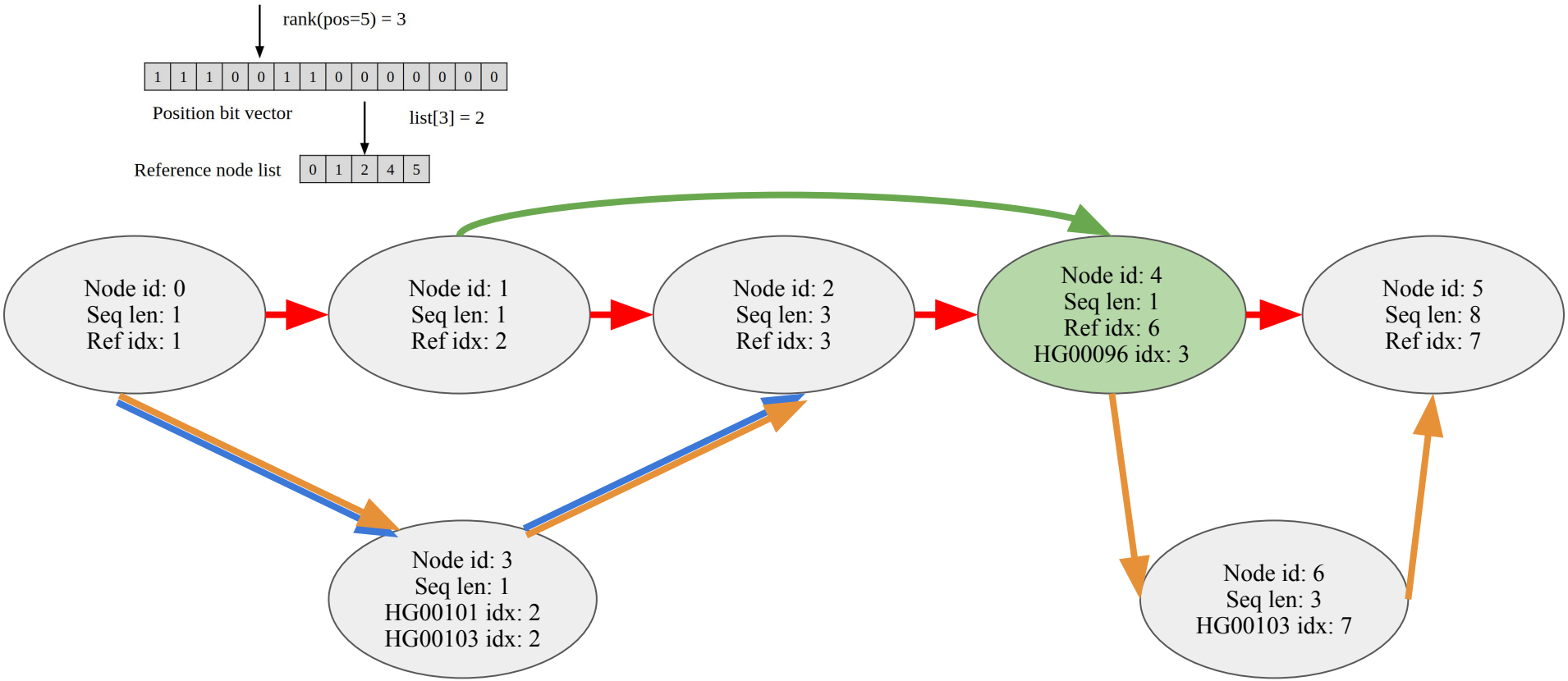
Example: Lookup sequence at pos 5 in HG00096



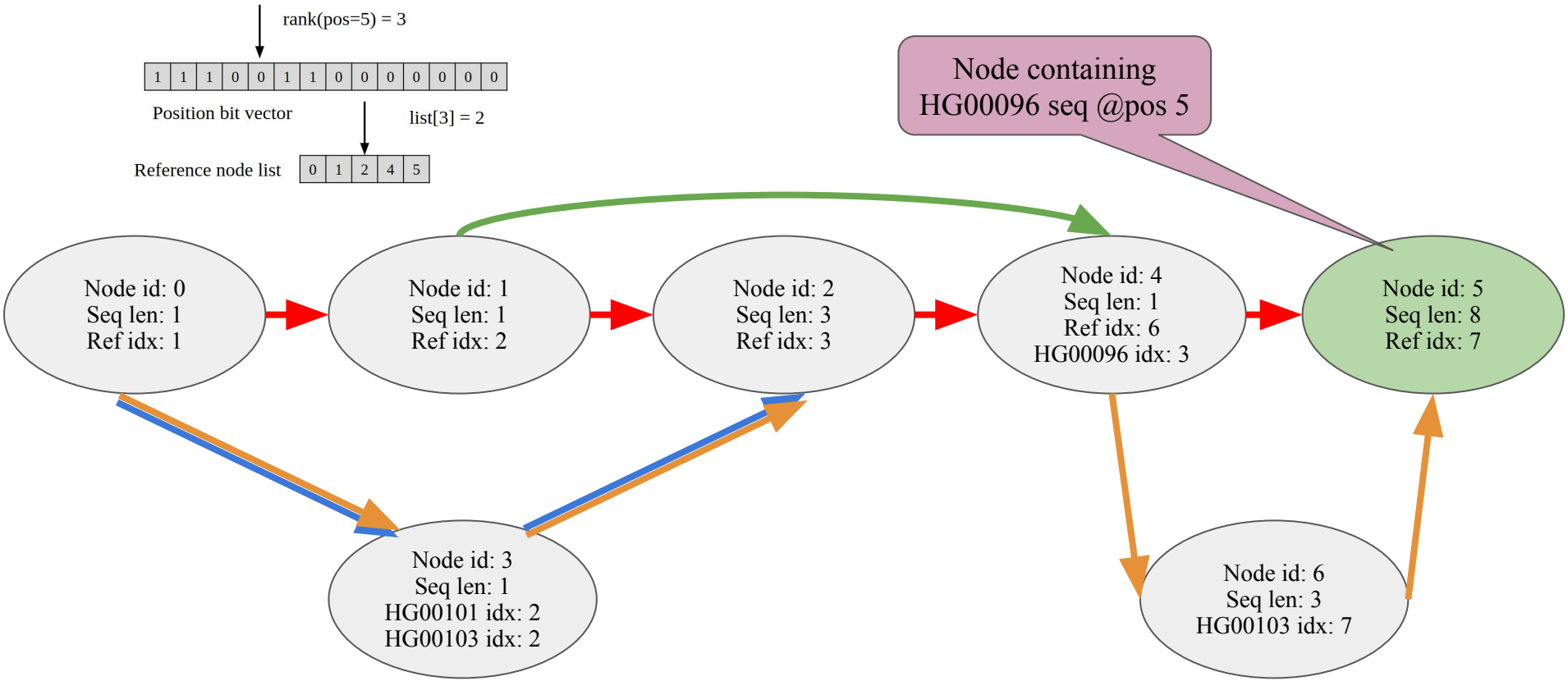
Example: Lookup sequence at pos 5 in HG00096



Example: Lookup sequence at pos 5 in HG00096



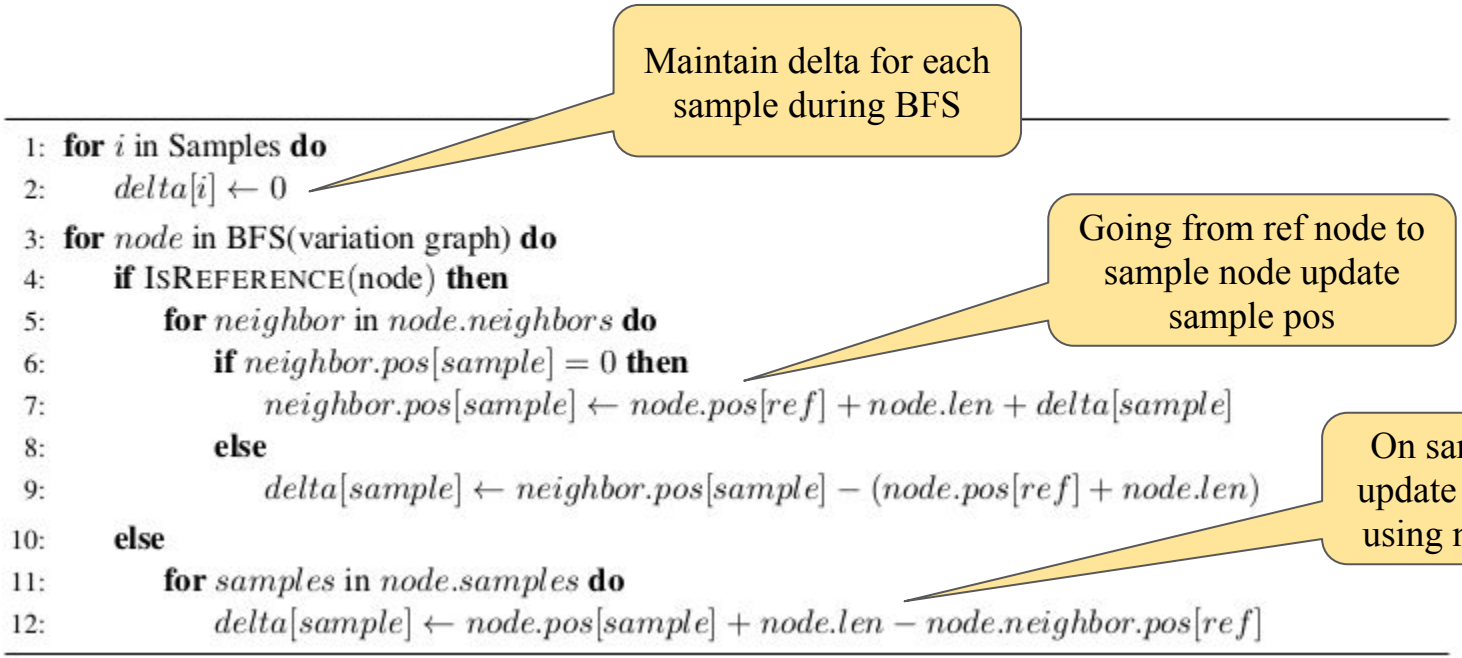
Example: Lookup sequence at pos 5 in HG00096



Two-phase construction avoids multiple backtracking during construction.

- Backtracking to compute absolute sample index is time consuming
- Backtracking during each variant addition becomes the bottleneck
- Phase 1: add variant by adding new nodes and splitting reference nodes
  - Do not assign any absolute index to sample nodes
- Phase 2: breadth-first traversal over the whole graph and update the absolute index for each sample

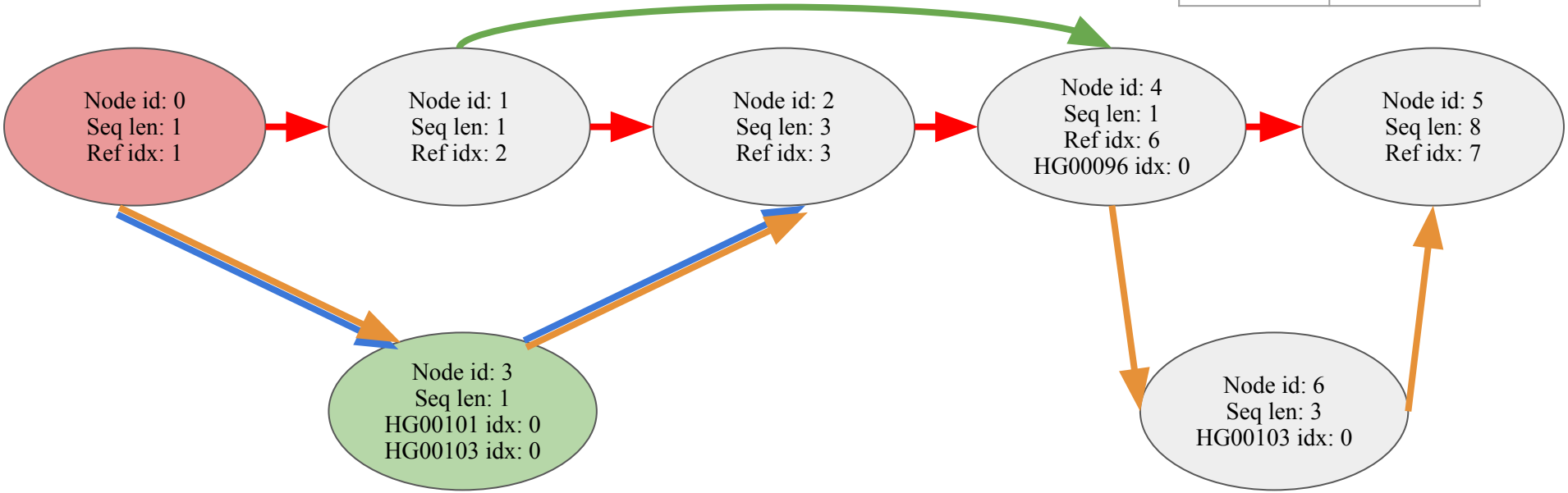
Pseudo code for updating sample indexes




```
1:  for i in Samples do
2:      delta[i] ← 0
3:  for node in BFS(variation graph) do
4:      if IsReference(node) then
5:          for neighbor in node.neighbors do
6:              if neighbor.pos[sample] = 0 then
7:                  neighbor.pos[sample] ← node.pos[ref] + node.len + delta[sample]
8:              else
9:                  delta[sample] ← neighbor.pos[sample] − (node.pos[ref] + node.len)
10:     else
11:         for samples in node.samples do
12:             delta[sample] ← node.pos[sample] + node.len − node.neighbor.pos[ref]
```

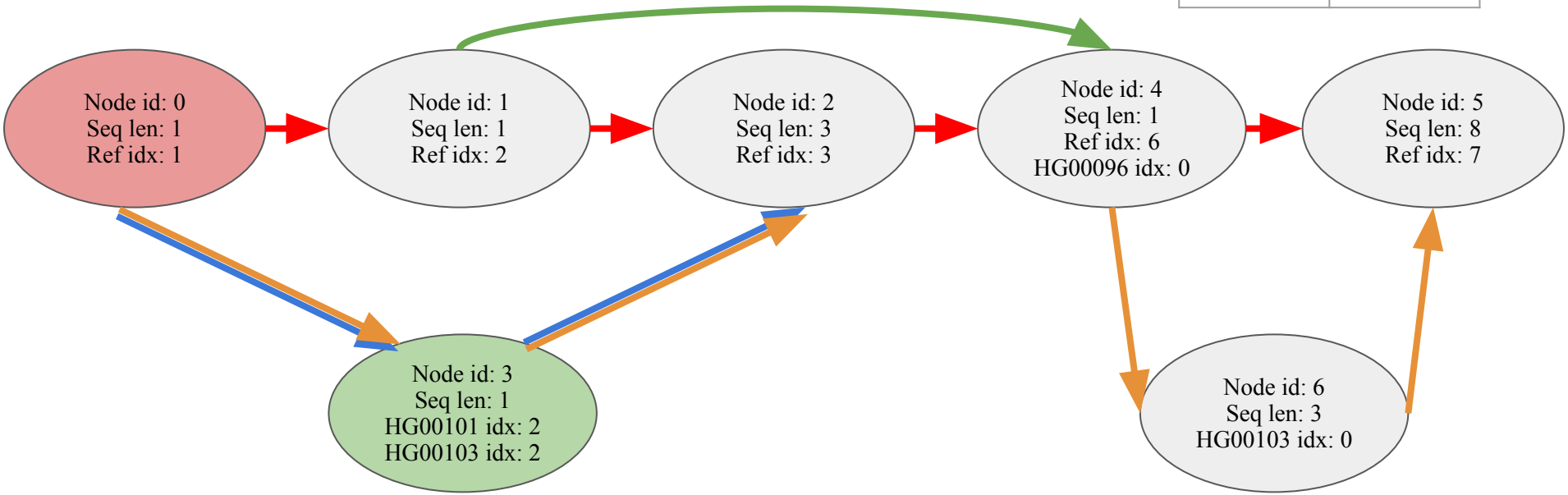
Phase 2: updating sample indexes

HG00096	**0**
HG00101	**0**
HG00103	**0**



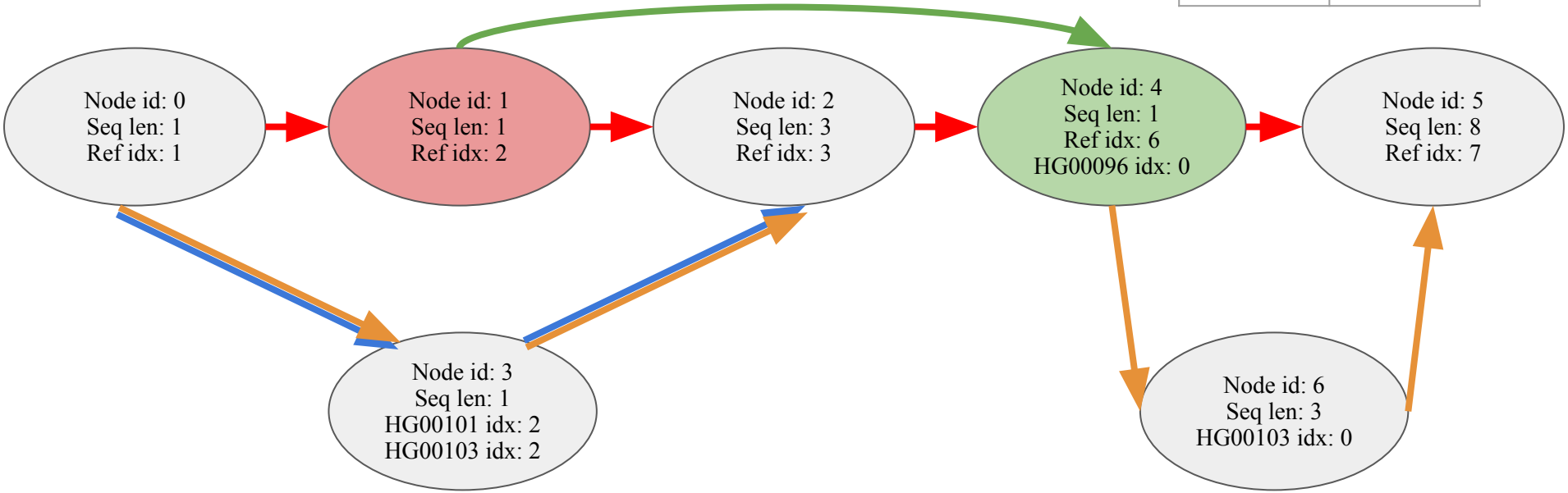
Phase 2: updating sample indexes

HG00096	**0**
HG00101	**0**
HG00103	**0**



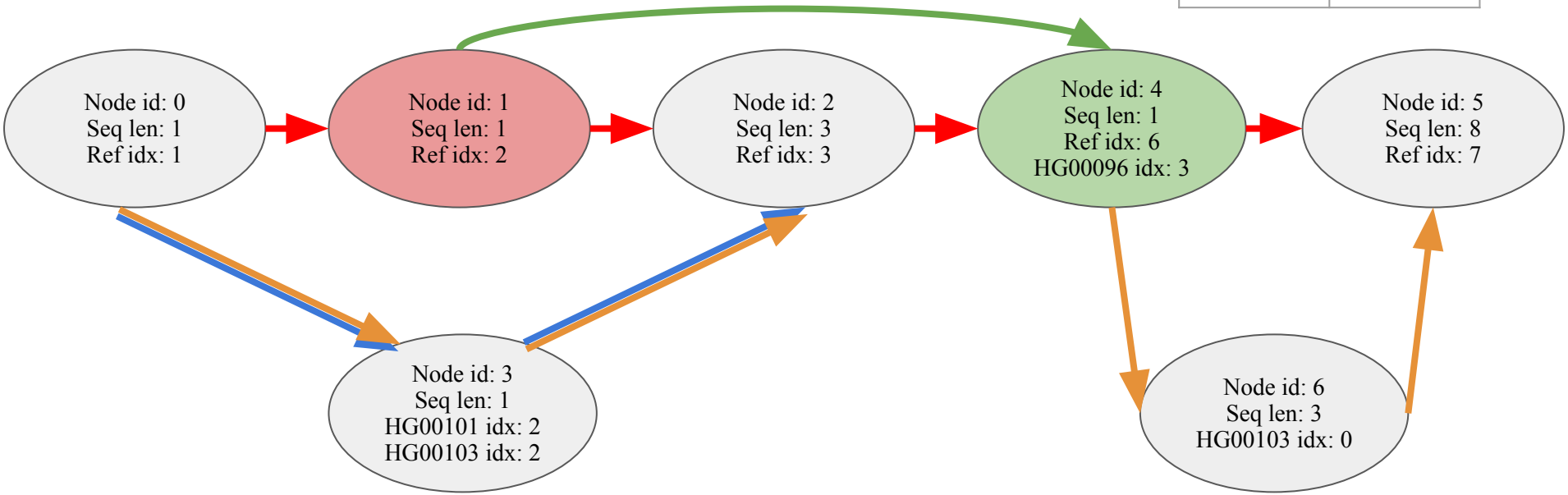
Phase 2: updating sample indexes

HG00096	**0**
HG00101	**0**
HG00103	**0**



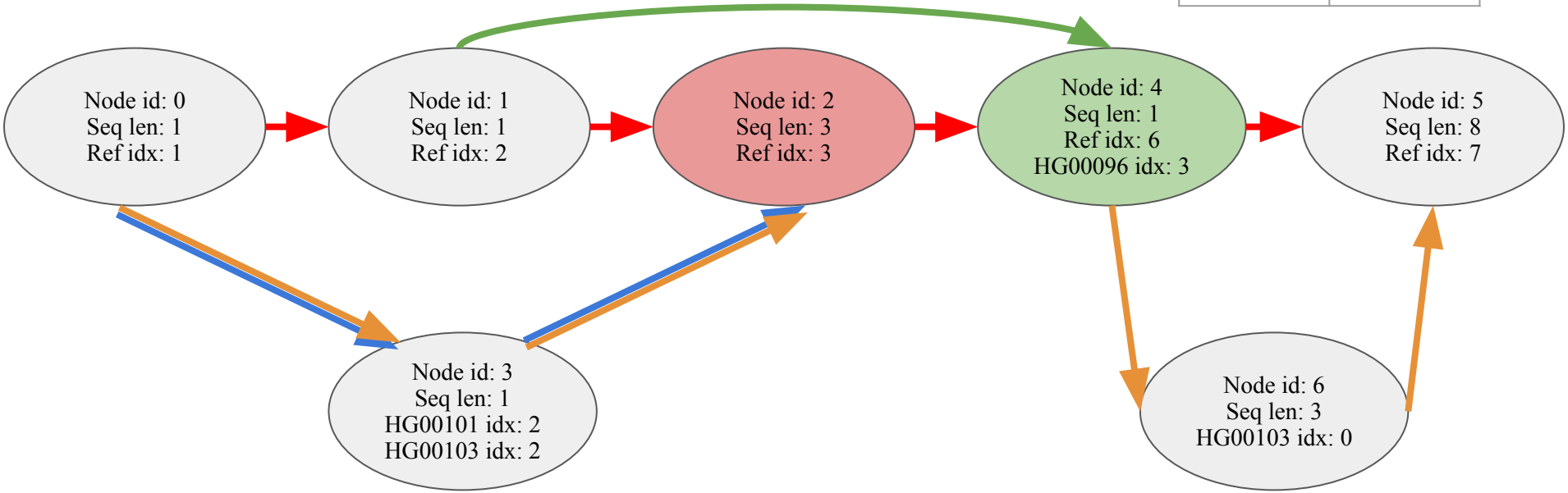
Phase 2: updating sample indexes

HG00096	**0**
HG00101	**0**
HG00103	**0**



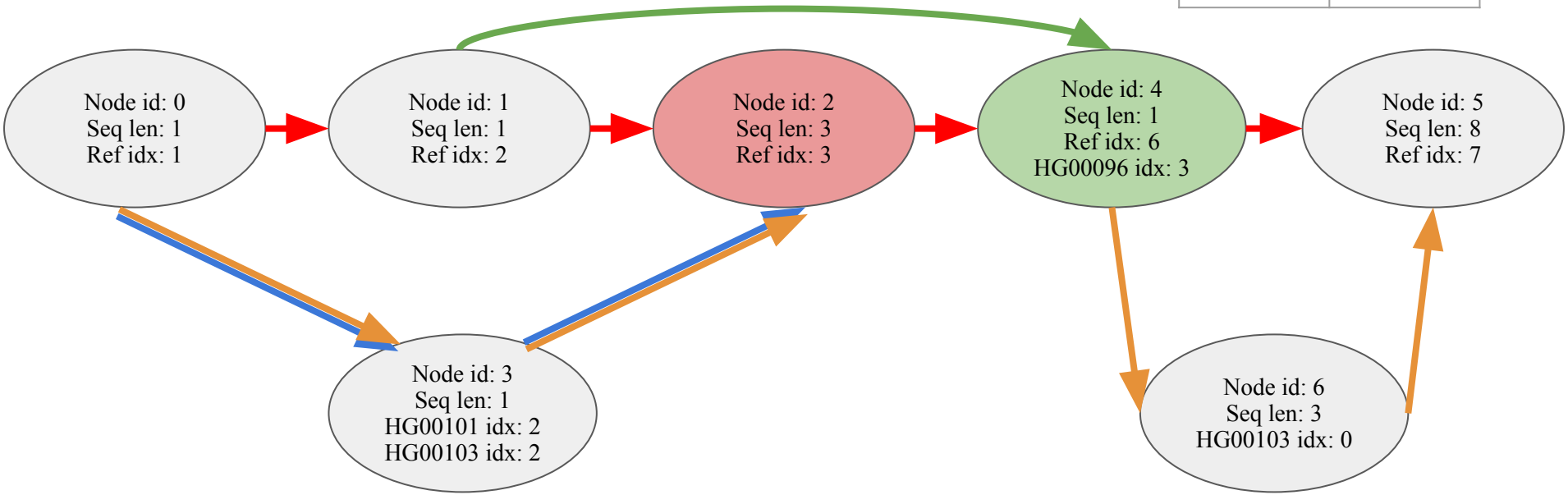
Phase 2: updating sample indexes

HG00096	**0**
HG00101	**0**
HG00103	**0**



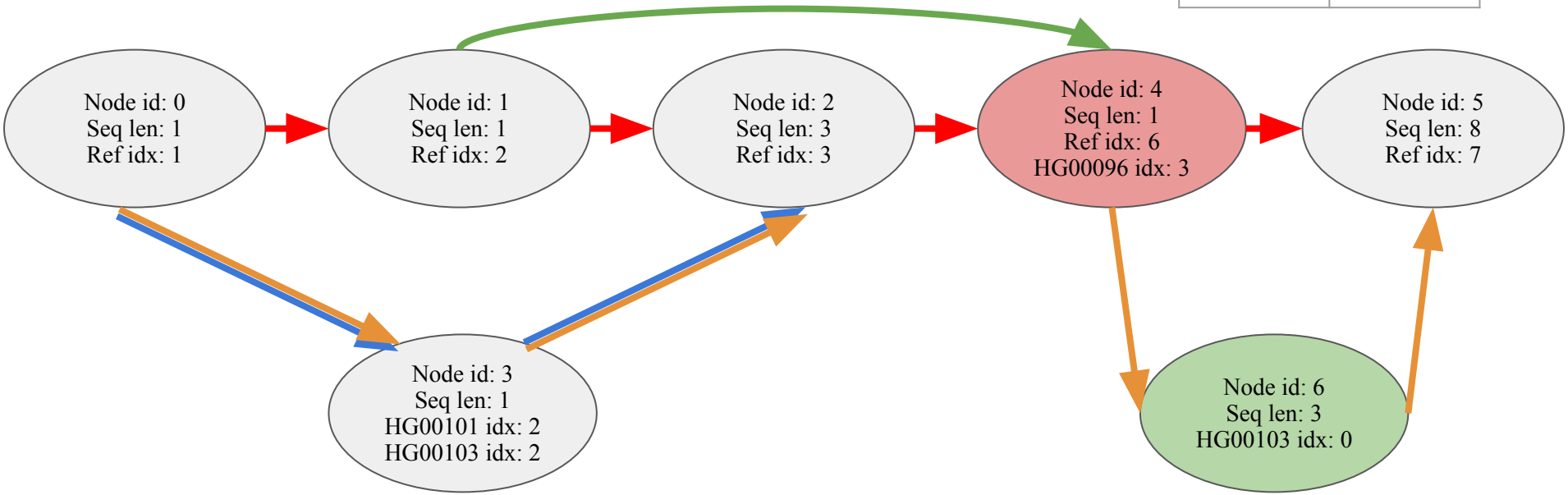
Phase 2: updating sample indexes

HG00096	**-3**
HG00101	**0**
HG00103	**0**



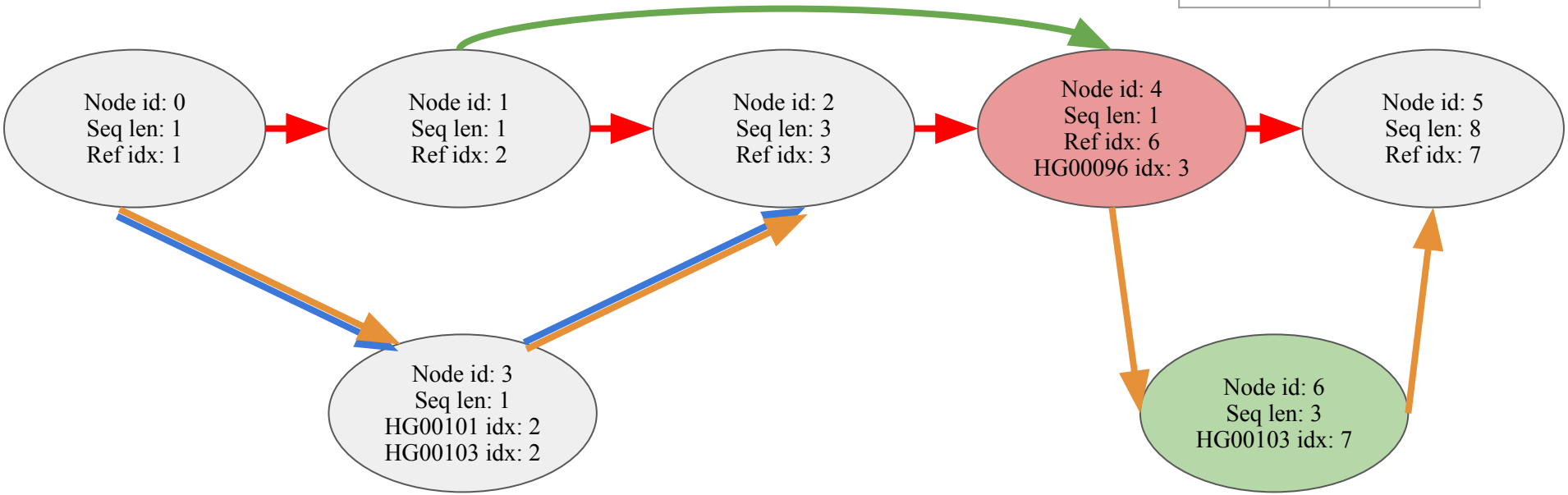
Phase 2: updating sample indexes

HG00096	**-3**
HG00101	**0**
HG00103	**0**



Phase 2: updating sample indexes

HG00096	**-3**
HG00101	**0**
HG00103	**0**



VariantStore components

- Variants are encoded in variation graph
- Position-based index to locate node in the graph corresponding to a position
- Position index is only maintained for a single reference genome
- Use local graph traversals to translate between coordinate systems
- Marker nodes in the graph store absolute sample positions and bound local traversals during translation
- **Graph partitioning to enable memory-efficient construction/query**

Graph partitioning for memory-efficient operations

- Variant queries are performed between small position ranges
- Queries only access a small portion of the graph
- Partition the graph into small chunks and only load relevant chunks during query
- Graph is partitioned dynamically during construction
- We sort queries based on position in a batch and perform them sequentially
- Each chunk is loaded only once during a batch operation

Performance evaluation

- Index construction performance
  - Running time
  - Disk space
  - Peak RAM
- Query performance
  - Running time
  - Peak RAM
  - Effect of range size and variant density
- Index space analysis

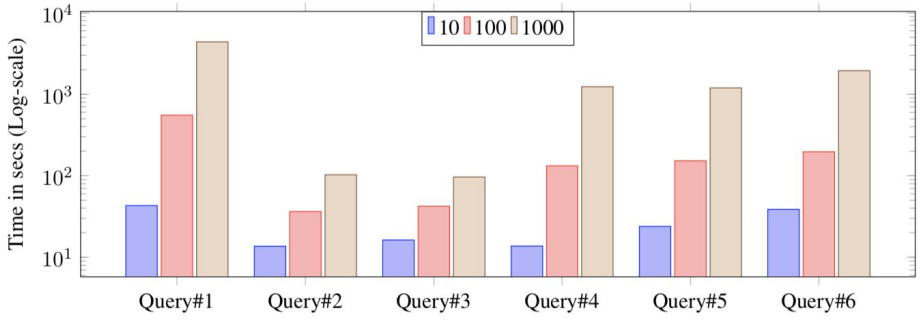
Results for constructing the index

System	Time	Disk space	Peak RAM	Peak RAM Agg.
Dataset		1000 Genomes		
VariantStore	3 Hrs 25 mins	41 GB	8.8 GB	153 GB
VG-toolkit	11 Hrs 10 mins	50 GB	37 GB	450 GB
Dataset		TCGA (OV)		
VariantStore	1 Hr 5 mins	3.4 GB	1.1 GB	17.45 GB
VG-toolkit		11 GB*		
Dataset		TCGA (LUAD)		
VariantStore	1 Hr 20 mins	3.5 GB	2.3 GB	36.05 GB
VG-toolkit		12 GB*		
Dataset		TCGA (BRCA)		
VariantStore	4 Hrs 36 mins	4.2 GB	3.2 GB	53.21 GB
VG-toolkit		14 GB*		

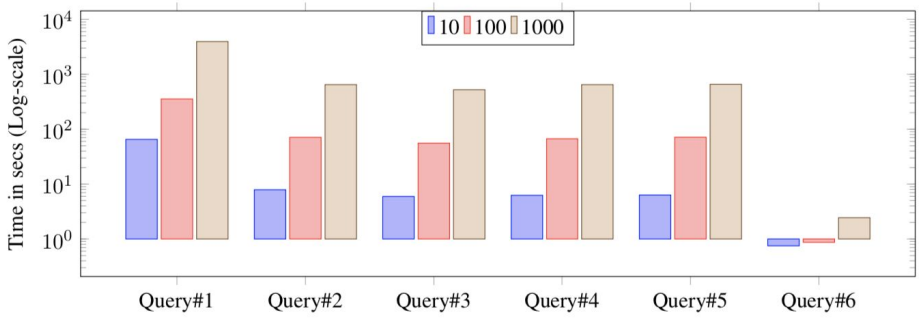
Table 1: Time, space, peak RAM, and peak RAM (aggregate) to construct variant index on the 1000 Genomes and TCGA (OV, LUAD, and BRCA) data using VariantStore and VG toolkit. *VG toolkit could not build GBWT index embedding all sample paths for TCGA data. Space reported is for the XG index that does not contain any path information. We constructed all 24 chromosomes (1 – 22 and X and Y) in parallel. The time and peak RAM reported is for the biggest chromosome (usually chromosome 1 or 2). The space reported is the total space on disk for all 24 chromosomes. The peak RAM (aggregate) is the aggregate peak RAM for all 24 processes.

VariantStore is 3× faster, takes 25% less disk space, and 3× less peak RAM than VG toolkit.

Results for variant queries



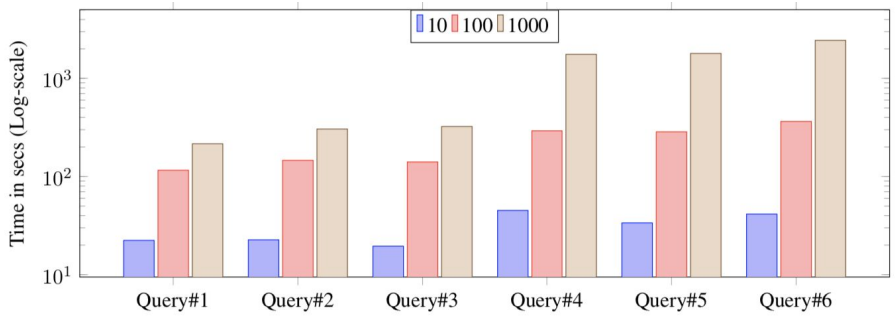
(a) Time for 10, 100, and 1000 queries on Chromosome 22 index in VariantStore for 1000 Genomes data.



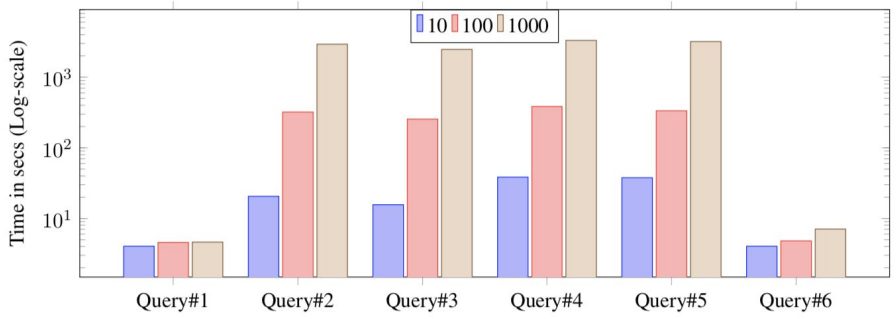
(b) Time for 10, 100, and 1000 queries on Chromosome 22 index in VariantStore for TCGA LUAD data.

Aggregate time to execute queries increases sublinearly with the number of queries

Results for variant queries



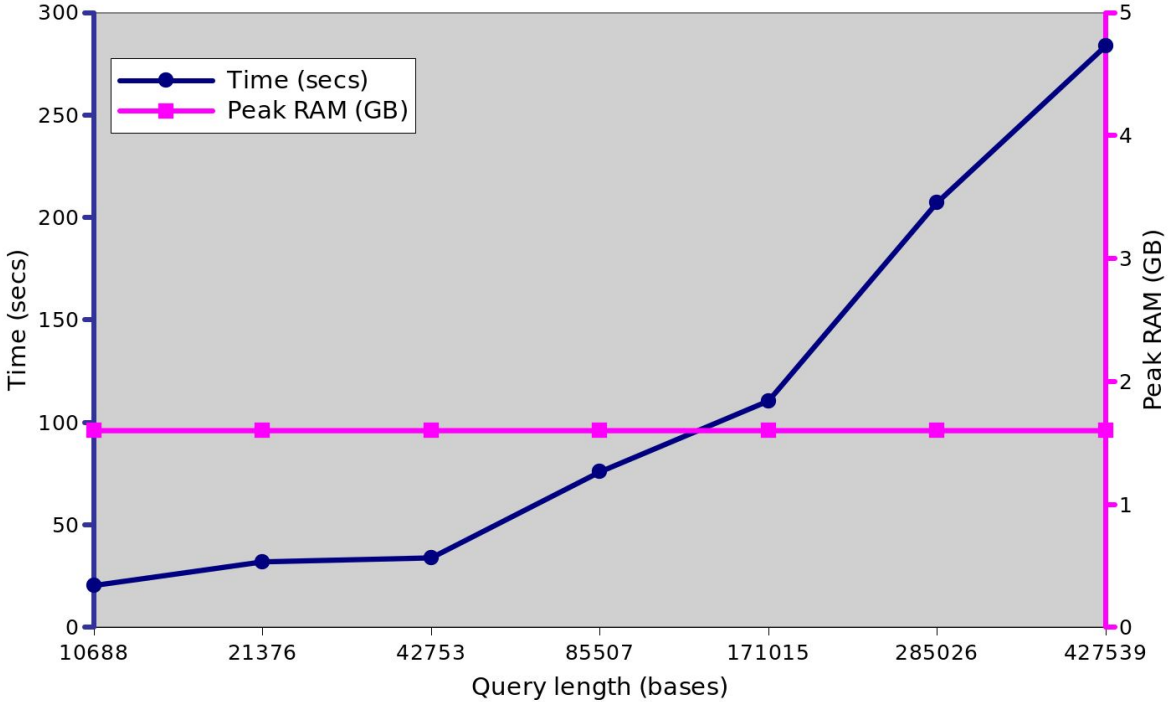
(a) Time for 10, 100, and 1000 queries on Chromosome 2 index in VariantStore for 1000 Genomes data.



(b) Time for 10, 100, and 1000 queries on Chromosome 2 index in VariantStore for TCGA LUAD data.

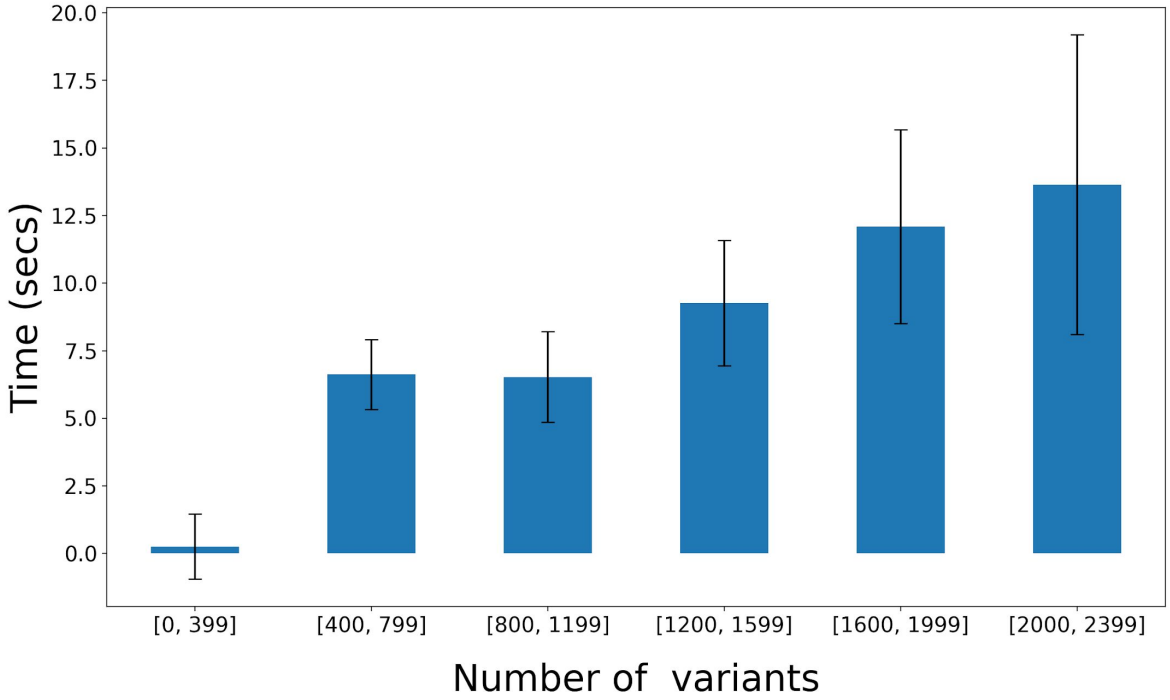
Takes between 0.002 -- 3 seconds for different types of variant queries

Query analysis based on range size



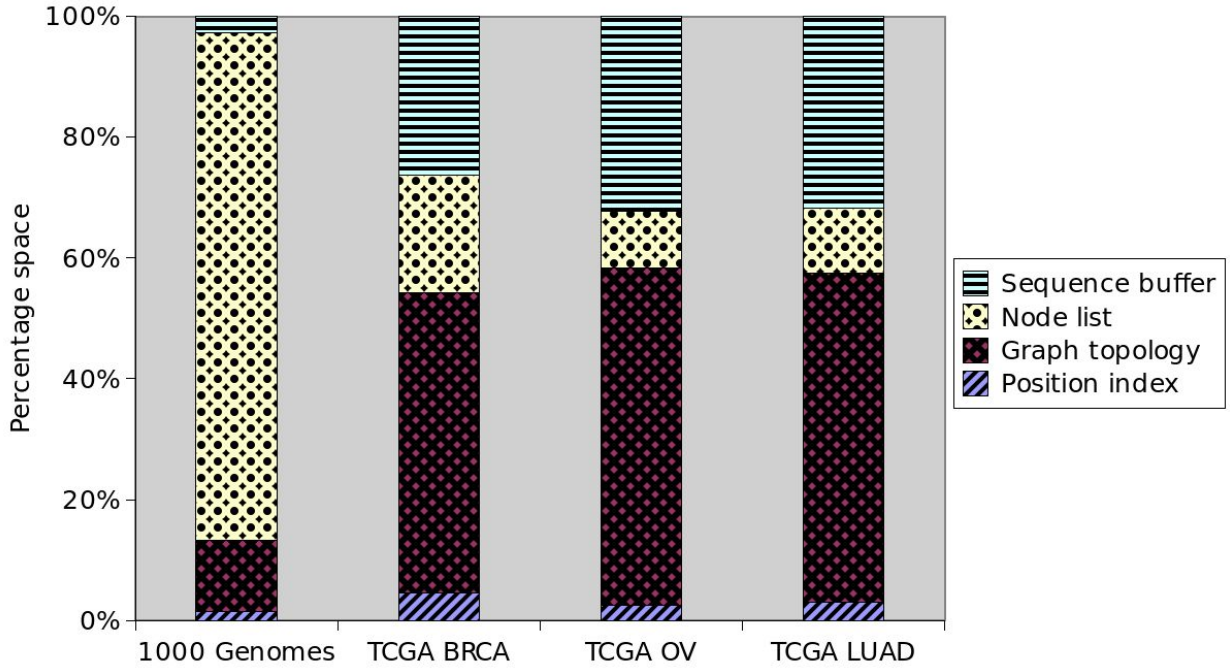
Memory usage remains constant regardless of the query length

Query analysis based on number of variants



Query time increases with the density of variants in the queries region

Index space analysis



Position index is only a small fraction (1.5%--4.5%) of the index size

Adding multiple references will have a trivial space overhead

Conclusion

- The ability to efficiently query population-level variation data promises to improve many medical and scientific applications
- VariantStore enables querying variants and sequences across thousands of samples in multiple coordinate systems
- https://www.biorxiv.org/content/10.1101/2019.12.24.888297v2