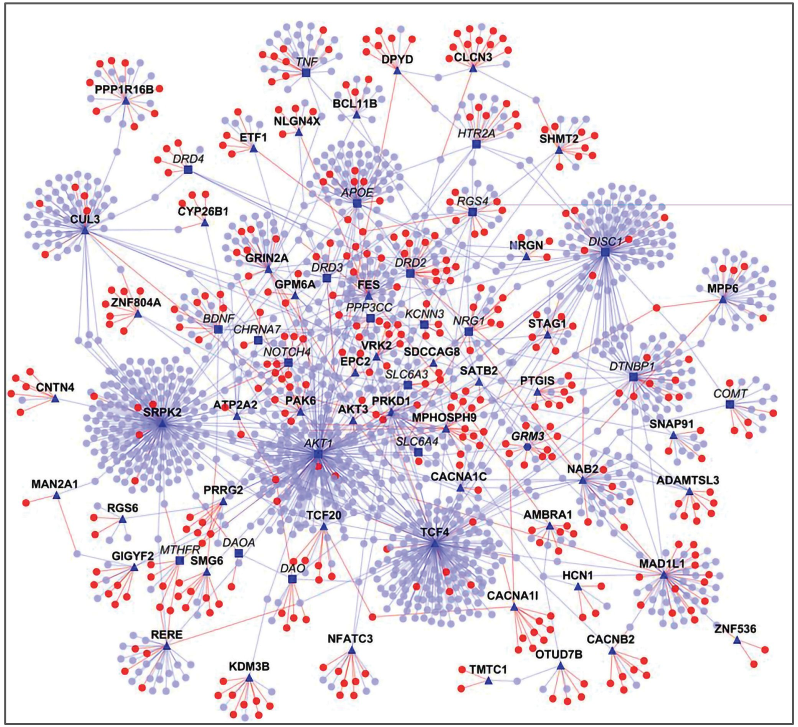
Population-Specific Multi-omics Graph Generation for a target protein

Group 7

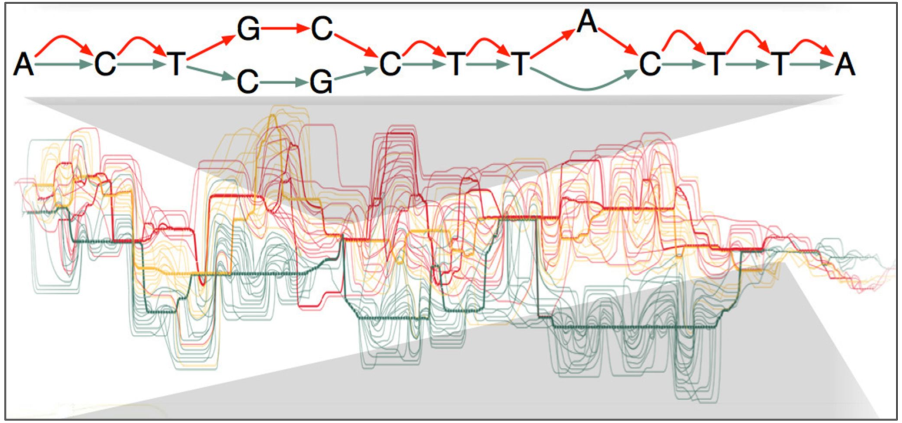
Siddharth Sabata, Shivank Sadasivan, Lars Warren Ericson, Arth Banka, Rachael Oluwakamiye Abolade

What we know..

Protein Network

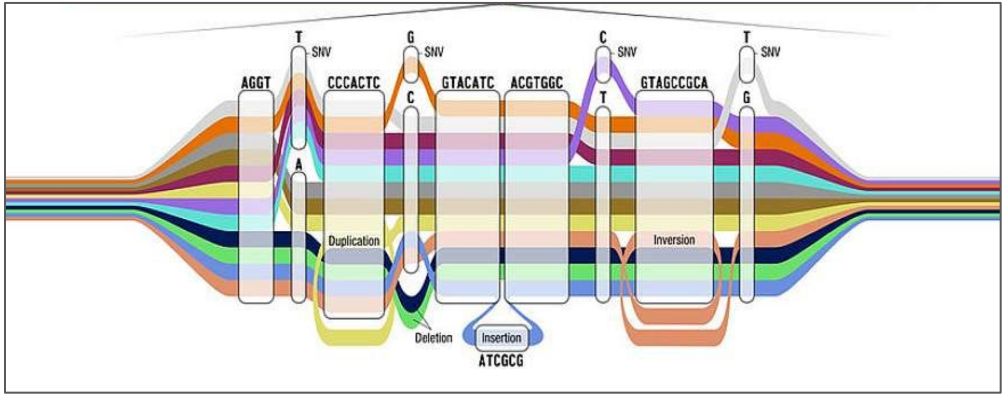


Genetic Variants in population



Goal :Integrate population specific pQTL information and population specific genetic variant information into one.

Population Gene Graphs



Variant Node:

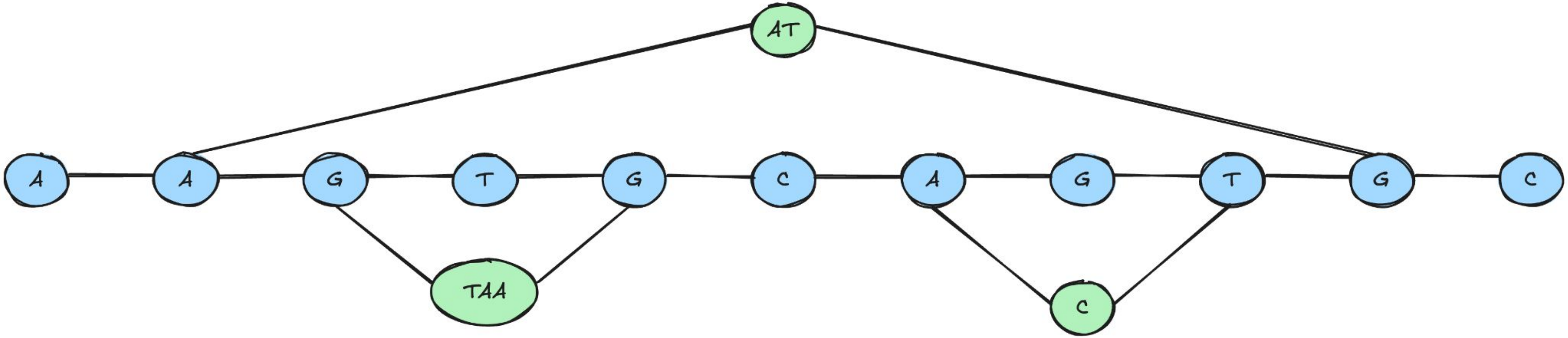
- Global genomic position
- Reference, alternate allele sequence, and their lengths
- Effect size of variant (beta)
- -log10 of the p-value
- Direction

Reference Node:

- Global genomic position
- Local index
- Nucleotide

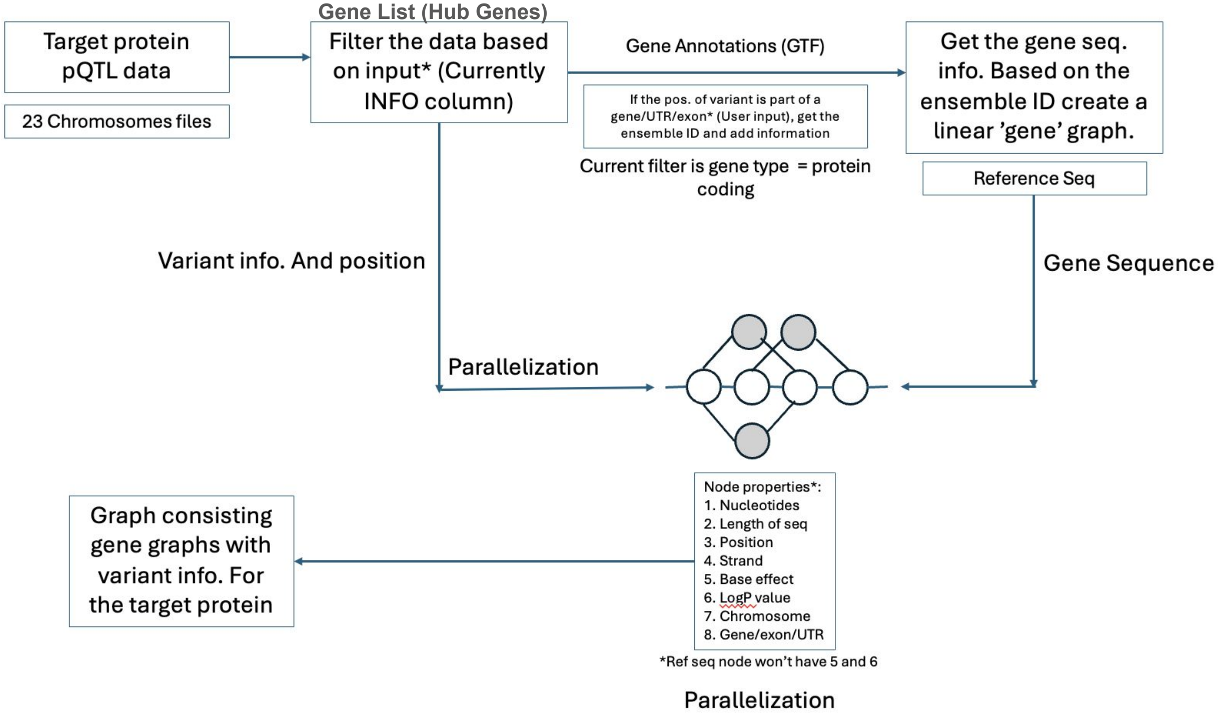
pQTL :Value that quantifies the genetic variations that influence the abundance of proteins in a sample, such as blood, tissue, or cell culture

Why graphs?



- Enables advanced graph-based analytics and machine learning applications
- Facilitates precision medicine by linking genomic variation with protein expression
- Provides a foundation for developing predictive models on variant impacts

Data integration and graph construction



Target protein pQTL data

23 Chromosomes files

Gene List (Hub Genes)

Filter the data based on input* (Currently INFO column)

Gene Annotations (GTF)

If the pos. of variant is part of a gene/UTR/exon* (User input), get the ensemble ID and add information

Current filter is gene type = protein coding

Get the gene seq. info. Based on the ensemble ID create a linear 'gene' graph.

Reference Seq

Variant info. And position

Gene Sequence

Parallelization

Node properties*:

1. Nucleotides
2. Length of seq
3. Position
4. Strand
5. Base effect
6. LogP value
7. Chromosome
8. Gene/exon/UTR

*Ref seq node won't have 5 and 6

Parallelization

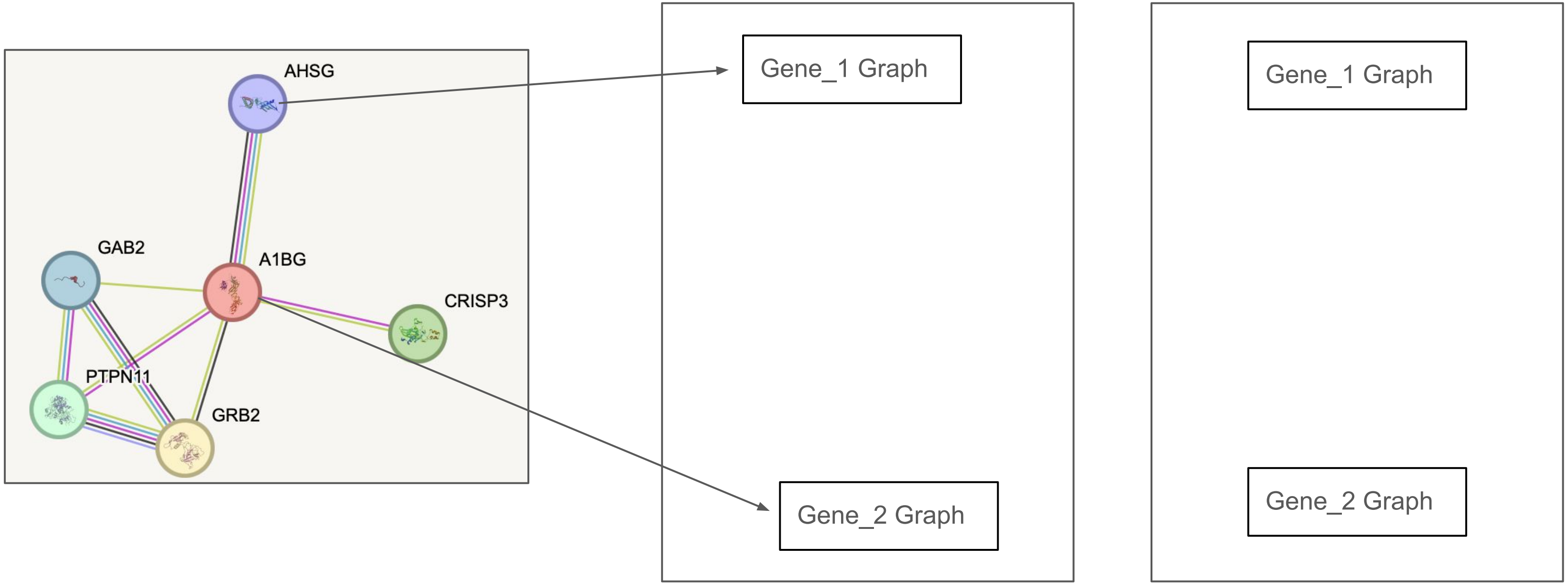
Graph consisting gene graphs with variant info. For the target protein

Build undirected graphs for individual genes using variant information and reference data

Leverage PyTorch Geometric (PyG) for efficient graph representation and downstream analysis

What are we doing?

Integrating structural variation and protein expression within a population genomics framework to generate gene-specific graphs for variant annotation and analysis



Gene_1 Graph

Gene_2 Graph

Population 1

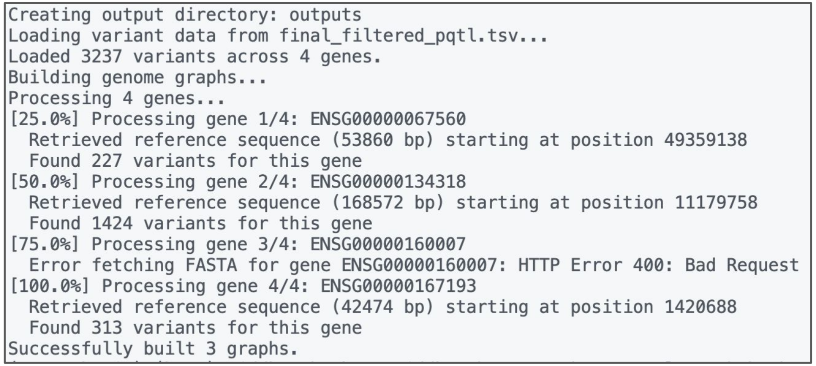
Gene_1 Graph

Gene_2 Graph

Population 2

Results and future work

Things we don’t control…



```
Creating output directory: outputs
Loading variant data from final_filtered_pqtl.tsv...
Loaded 3237 variants across 4 genes.
Building genome graphs...
Processing 4 genes...
[25.0%] Processing gene 1/4: ENSG00000067560
  Retrieved reference sequence (53860 bp) starting at position 49359138
  Found 227 variants for this gene
[50.0%] Processing gene 2/4: ENSG00000134318
  Retrieved reference sequence (168572 bp) starting at position 11179758
  Found 1424 variants for this gene
[75.0%] Processing gene 3/4: ENSG00000160007
  Error fetching FASTA for gene ENSG00000160007: HTTP Error 400: Bad Request
[100.0%] Processing gene 4/4: ENSG00000167193
  Retrieved reference sequence (42474 bp) starting at position 1420688
  Found 313 variants for this gene
Successfully built 3 graphs.
```

Results

- Developed a workflow given pQTL , Genes and GTF files.
- Successfully generated compressed graph file
- Used test dataset from (AOB protein) pQTL data for American population

Future work

- **Parallelized Graph Generation** – Distribute graph construction across multiple genes for faster processing.
- **Advanced Filtering** – Implement filters for gene lists, features, gene types, and coordinates for problem-specific graphs.
- **GNN Integration** – Utilize Graph Neural Networks for variant annotation and downstream analysis.
- **Scalability & Optimization** – Enhance computational efficiency and adaptability for large-scale genomic studies.