BioHackrXiv

Addressing Background Genomic and Environmental Effects on Health through Accelerated Computing and Machine Learning: Results from the 2025 Hackathon at Carnegie Mellon University

Siddharth Sabata [1], Jędrzej Kubica [2], Rishika Gupta [3], Lars Warren Ericson [4], Halimat Chisom Atanda [5], Gobikrishnan Subramaniam [6], Abraham G. Moller[7], Rachael Oluwakamiye Abolade [8], Arth Banka [1], Samuel Blechman [9], Rorry Brenner[10], Maria Chikina[9], Li Chuin Chong [11], Nicholas P. Cooley [9], Daniel Chang [1], Phil Greer [12], Anshika Gupta[9], Avish A. Jha [1], Emrah Kacar[13], Nanami Kubota[9], William Lu[1], Louison Luo[1], Tien Ly [14], Rajarshi Mondal[15], Ciara O'Donoghue [13], Aung Myat Phyo [16], Peng Qiu[1], Glenn Ross-Dolan[17], Ali Saadat [18], Shivank Sadasivan [1], Rebecca Satterwhite[9], Soham Shirolkar[19], Yuning Zheng[1], Huajin Wang [1], Melanie Gainey [1], and Ben Busby[20]

1 Carnegie Mellon University, Pittsburgh, Pennsylvania **2** Univ. Grenoble Alpes, CNRS, UMR 5525, TIMC / BCM, 38000 Grenoble, France **3** CiTIUS ~ Centro Singular de Investigación en Tecnoloxías Intelixentes, Santiago de Compostela, Spain **4** Catskills Research Company, Huntersville, North Carolina **5** Mater Research Institute-The University of Queensland, Queensland, Australia **6** Queen's University Belfast, Belfast, United Kingdom **7** National Institute on Aging (NIA) Center for Alzheimer's and Related Dementias(CARD), Bethesda, Maryland **8** Information Science Department, University of Arkansas at Little Rock, Pharmaceutical Science Department, University of Arkansas for Medical Sciences **9** University of Pittsburgh, Pittsburgh, Pennsylvania **10** Perforated AI, Pittsburgh, Pennsylvania **11** Institute for Experimental Virology, TWINCORE Centre for Experimental and Clinical Infection Research, a Medical School Hannover (MHH) and Helmholtz Centre for Infection Research (HZI) joint venture, Hannover, Germany **12** Ariel Precision Medicine, Pittsburgh, Pennsylvania **13** Complex Trait Genetics Lab, Smurfit Institute of Genetics, Trinity College Dublin, Ireland **14** Department of Computer Science, College of Science, San Jose State University, San Jose, California **15** Department of Bioinformatics, Pondicherry University, India **16** Family Genomics Research Group, Department of Biology, Maynooth University, Ireland **17** APC Microbiome Ireland, University College Cork, Ireland **18** School of Life Sciences, EPFL, Lausanne, Switzerland **19** University of South Florida, Tampa, Florida **20** DNAnexus, Current Address NVIDIA

BioHackathon series:
CMU-DNAnexus Collaborative Bioinformatics Hackathon
Pittsburgh, PA, 2025
BioHackrXiv

Submitted: 03 Jun 2025

Published by BioHackrXiv.org

1. Introduction

Overall Event

In March 2025, 34 scientists from the United States, Ireland, the United Kingdom, Switzerland, France, Germany, Spain, India, and Australia gathered in Pittsburgh, Pennsylvania and virtually for a collaborative biohackathon, hosted by DNAnexus and Carnegie Mellon University Libraries. The goal of the hackathon was to explore machine learning approaches for multimodal problems in computational biology using public datasets. Teams worked on the following innovative projects: applying machine learning techniques for clustering and similarity analysis of haplotypes; adapting the StructLMM framework to study Gene-Gene (GxG) interactions; creating a nextflow workflow for generating an imputation reference panel using large-scale cohort data; optimizing discovery of causal relationships in large electronic health record (EHR)

datasets using the open source causal analysis software Tetrad; examining the evolution of a graph neural network in a Lenski-esque experiment; and developing tools and workflows for generating pathway intersection diagrams and graph-based analyses for multiomics data. All projects were dedicated to study the background genomic and environmental effects underlying complex genotype-phenotype relationships. Their objective was to set foundations for further studies on predicting complex phenotypic traits using integrative multi-omic and environmental analyses. All team projects are detailed below:

1.1 Clustering of haplotype matrices

Haplotype analysis plays a critical role in understanding genetic variation and evolutionary relationships. This study presents a computational pipeline on DNANexus that integrates haplotype data processing, ARG reconstruction, and machine learning techniques to explore genetic similarity and clustering among human samples. We used SHAPEIT2 phased variant call format (VCF) files from chromosomes 6, 8, 21, and 22 of The 1000 Genomes Project (Consortium, 2015), converted the data into haplotype (HAP) format using Plink2 (Chang et al., 2015; Purcell & Chang, n.d.) and applied preprocessing steps to standardize the input for ARG Needle. We also filtered chromosome 6 haplotypes for TNF and HLA-A variants and chromosome 8 for beta defensin, as TNF is one of the least variable genes in the human genome, while HLA-A and beta defensin are amongst the most variable. We obtained 61, 313, and 486 deduplicated biallelic SNPs for TNF, HLA-A, and beta-defensin, respectively. We then performed hierarchical clustering and similarity matrix calculation from these gene-specific haplotypes.

1.2 Cis and trans effects of haplotypes on rare variants penetrance with StructLMM adapted for Gene-Gene interaction analysis

Gene-gene (GxG) interactions play a crucial role in understanding complex traits and diseases, yet their detection can be challenging due to statistical power limitations and confounding factors. Traditional genome-wide association studies (GWAS) primarily focus on single-locus effects, often overlooking interactions between genetic variants (Cordell, 2009; Hu et al., 2014). Linear mixed models (LMM) have been widely used to account for population structure and relatedness, yet detecting GxG interactions remains underexplored (Alamin et al., 2022). StructLMM, an LMM introduced by (Moore et al., 2018), has been successfully applied to gene-environment (GxE) interactions by incorporating structured environmental effects.

Here, we adapt StructLMM to detect GxG interactions by leveraging local ancestry principal components (PCs), which capture variation at a haplotype level, as a proxy for environmental influences. Integrating ancestry-informed structure into the model allows the detection of GxG interactions while accounting for population heterogeneity. Such an approach provides a flexible and scalable framework for studying interactions across different genomic regions.

1.3 Generation of imputation panels for combined sequencing with biobank data

Blended Genome Exome (BGE) sequencing is an innovative approach developed by the Broad Institute that integrates low-pass whole genome sequencing (WGS) at approximately 3x coverage with 30x coverage whole exome sequencing (WES) on a unified sequencing platform (DeFelice et al., 2024). Unlike traditional genotyping arrays, BGE and low-pass WGS are not limited by predefined probe sets based on specific ancestral data, making them more inclusive for diverse populations. To obtain accurate variant calls for common variants across the genome, both BGE and low-pass WGS data require an imputation step, which relies on high-quality reference panels comprising large, ancestrally diverse samples. In this project, we aim to develop a Nextflow workflow (Di Tommaso et al., 2017) to construct imputation reference panels using extensive cohort datasets. As a proof of concept, we will deploy this

workflow on the combined 1000 Genomes Project (1kGP) (Consortium, 2015) and The Human Genome Diversity Project (HGDP) dataset (Cavalli-Sforza, 2005) made available through gnomAD (Koenig et al., 2023).

1.4 Rapid Longitudinal Analysis of Public Health Data

The increasing availability of electronic health records (EHRs) has revolutionized medical research, enabling large-scale data-driven insights into patient outcomes, disease progression, and treatment effectiveness. The MIMIC-III v1.4 (Medical Information Mart for Intensive Care) dataset is one of the most widely used publicly available ICU (Intensive Care Unit) datasets, containing data of over 40,000 patients (Computational Physiology, 2016; Goldberger et al., 2000; Alistair Johnson et al., 2016; A. E. W. Johnson et al., 2016). Discovering causal relationships among clinical variables remains difficult. Working with raw EHR data presents several challenges that must be addressed for effective causal discovery: 1) data fragmentation is a significant issue, as of 27 interrelated tables MIMIC-III v1.4 require extensive preprocessing and integration; 2) high dimensionality poses computational challenges, with thousands of variables that need to be processed, filtered, and analyzed to extract relevant causal relationships; 3) the presence of noisy and missing data, due to irregular sampling and inconsistent documentation, which is common in real-world ICU settings; 4) large-scale causal searches demand efficient data storage solutions and scalable computational resources. These data inconsistencies can introduce biases and reduce reliability of causal inference models.

1.5 Lenski-esque GNN Competition Trials

Genomic medicine seeks to uncover molecular mechanisms responsible for human diseases. Experimental identification of novel disease-associated genomic variants is expensive and time-consuming, often requiring extensive clinical studies. Large biological networks provide crucial information on complex relationships and interactions between biomolecules (e.g., genes or proteins) that underlie human diseases (Barabási et al., 2011). Network-based computational methods provide an opportunity to efficiently model these complex relationships. In this study, we leveraged a graph neural network for disease-gene prioritization, geneDRAGGN (A. Altabaa et al., 2022), to perform a Lenski-esque experiment of "evolving" neural networks (Lenski, 2001). We aimed at studying how the performance of neural networks change when constructed using different architectures (i.e., different combinations of hidden layers). We started with the original architecture of geneDRAGNN, then we iteratively "evolved" the neural network; in each iteration, we replaced individual layers of the network with various combinations of different layers. We aimed at selecting combinations of layers that have the highest impact on the neural network performance in discovery of novel disease-associated genes.

1.6 Population-Specific Multiomics Graph Analysis of ACE Protein Expression

Graph-based models offer a powerful approach for integrating multi-omics data to study gene regulation and protein expression. Recent advances in graph attention networks have demonstrated superior performance in cancer classification by effectively capturing complex molecular relationships (Alharbi et al., 2025). Additionally, specialized applications like SSGATE have extended graph-based approaches to both single-cell and spatial multi-omics integration through dual-path graph attention auto-encoders, enabling more comprehensive analysis of cellular heterogeneity across different tissue types and sequencing technologies (Lv et al., 2024). This study presents a graph-based multi-omics framework that combines protein quantitative trait loci (pQTL) data, genome annotations (GTF files), and the GRCh38.p14 reference genome to map genetic variants affecting protein expression systematically (CRG, 2021; Wang et al., 2024). Our method constructs population-specific genome graphs, where subgraphs represent gene-level regulatory interactions, incorporating variant effects, statistical significance, and functional annotations. This structured representation enables the identification of cis-

and trans-acting variants, uncovering regulatory differences across populations. By utilizing graph-based modeling, our framework enhances the interpretation of genetic influences on protein expression, providing a scalable and integrative tool for multi-omics analysis, precision medicine, and systems biology.

2. Methods

2.1 Clustering of haplotype matrices

In this study, we developed a pipeline to analyze haplotype data from The 1000 Genomes Project (https://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/) and apply machine learning techniques for clustering and similarity analysis. The methodology is outlined as follows:

Step 1: Getting the Data

We utilized phased variant call format (VCF) files for chromosomes 6, 8, 21, and 22 from The 1000 Genomes Project (Consortium, 2015). These VCF files were pre-phased using SHAPEIT2 /cite{Delaneau2013}. We selected chromosome 6 as it was used by prior groups and contains HLA and chromosome 8 as it contains beta defensin, a highly variable gene involved in microbial immune response, and chromosomes 21/22 due to their smaller sizes allowing for test processing.

Step 2: Converting the Data

The VCF files were converted into haplotype (HAP) format using Plink2 (Chang et al., 2015; Purcell & Chang, n.d.).

Step 3: Preprocessing the Data

The HAP files were preprocessed with the following steps:

- Space delimitation was enforced as required by ARG-Needle.
- In the .sample files, the IDs in columns ID_1 and ID_2 were made identical via copying ID_2 to ID_1.
- In the .haps file, unique IDs were assigned to variants with missing identifiers.
- The maximum allele length was set to 280 to standardize input data.
- Combining it with columns 2-4 from the original sample file to create a new SNP name/ID.
- Creating a new sample file with the modified format.

Step 4: Generating ARGs

To additionally prepare the data for generation of ARGs, both map and hap files needed to be modified so that positions to be arranged in monotonically increasing order by removing duplicated variants. In accordance with Zhang et al. (Zhang et al., 2023), we performed ARG inference in parallel by dividing phased data into equal, non-overlapping chunks, and performing ARG inference on each chunk.

Step 5: Clustering Analysis and Visualization

Clustering/unsupervised machine learning pipelines were initially established using the ARGN file generated from the example SNP data described in the ARG-Needle \{argneedle}.

A tree visualization method for the ARGs that extends the efforts of prior hackathon teams was similarly initially produced from the ARG-Needle example SNP data using tskit.

We also ran clustering analysis (hierarchical clustering) on haplotype data for beta defensin, TNF alpha, and HLA-A genes from chromosomes 8 and 6 (both TNF alpha and HLA-A), respectively. We performed hierarchical clustering on input beta-defensin, TNF alpha, and HLA-A Plink hap files both by 1000 Genomes variant and individual using the seaborn clustmap function.

To characterize the haplotypes based on clinical significance, we used the vcf files to isolate biallelic SNPs, and then wrote code to use OpenCravat to annotate chromosomes with ClinVar ACMG annotation.

2.2 Cis and trans effects of haplotypes on rare variants penetrance with StructLMM adapted for Gene-Gene interaction analysis

Our approach adapts the StructLMM framework to enable the detection of gene-gene (GxG) interactions by replacing the traditional environment matrix with local ancestry PCs. This implementation allows us to model GxG interactions in an ancestry-aware manner, accounting for population-specific effects in the detection of interaction in any genomic regions of interest (cis or trans) while adjusting for confounders.

The workflow consists of three primary steps:

1. **Defining the query variant**: A single nucleotide variant (SNV) with a strong effect on a phenotype is selected as the primary genetic factor of interest.

2. **Extracting local ancestry PCs**: We compute principal components from local ancestry tracts within a genomic region of interest, which serve as structured covariates in the model.

3. **Applying StructLMM for interaction testing**: The modified StructLMM framework models the interaction between the query SNV and other variants in the specified genomic region, while local ancestry PCs control for population-specific effects.

Mathematical model:
The adapted model is structured as:

$$y = M\alpha + g\beta_0 + g\beta_1 + e + \epsilon$$

where y represents the phenotype vector, containing the observed trait values for the individuals, M is a matrix of covariates that includes any relevant fixed effects or confounders, g denotes the genotype of the query single nucleotide variant (SNV), which is selected based on its strong effect on the phenotype of interest, 0 is the parameter associated with the main effect of query SNV on the phenotype,

Methods – Operation (how do people use it?)

Users can follow the example use code provided in the GitHub repository (https://github.com/collaborativebioinformatics/Cis_and_trans_effects_on_variant_penetrance?tab=readme-ov-file#usage). The processes encoded consist of the following analyses:

Extracting local ancestry PCs and the single variant of interest:

- Variants from the genomic region of interest are filtered from a VCF file.

- Quality control steps are applied. This includes missingness filtering, Hardy-Weinberg equilibrium checks, minimum allele frequency, maximum allele counts, variant IDs handling, and pruning out variant pairs of high correlation (>0.2).

- Principal components (PCs) are computed to summarise ancestry variation.

- The processed SNV file was a two-column CSV with individual IDs and genotype values (0/1/2), and the PC file followed the standard output format from PCA tools (sample IDs + PC columns)

Running StructLMM for GxG analysis:

- The query SNV and local ancestry PCs are input into StructLMM.
- The phenotype and additional covariates (e.g., plink format phenotype data) are included in the model.
- Interaction effects are estimated, and statistical significance between the genetic variant and the PCs is provided to the user.

To evaluate the feasibility of our approach, we tested the modified StructLMM framework on the HAPNEST synthetic dataset (Wharrie et al., 2022). The full dataset includes genotypes for over one million individuals and 9 continuous phenotypic traits. For demonstration, we used the provided example subset comprising 600 individuals, Plink genotype files (.bed/.bim/.fam), local ancestry information (.sample file), and 9 phenotype files in .pheno format.

We selected a query SNV (chr6:32529369C>A; rs554894601) and computed local ancestry principal components (PCs) from a nearby region (chr6:29944513–29945558). This region and variant were chosen as illustrative input, although the region overlaps the Human gene HLA and the variant is intronic with no known clinical significance. This setup allows us to validate the tool's operation in a realistic workflow involving real variant data, regional ancestry structure, and phenotypic traits. From the nine available phenotypes, we selected phenotype 1 (.pheno1) with heritability of 0.03 and polygenicity of 0.1 and incorporated it as a binary trait, labeled Phenotype(binary) (Wharrie et al., 2022).

The StructLMM pipeline completed successfully, with the following output:

P-value	N (samples)	PCs used	Phenotype	Mean	Std Dev
0.861	600	5	Phenotype(binary)	0.502	0.500

No significant interaction was observed between the SNV and the local ancestry PCs ($p = 0.86$). This might be explained by the small sample size (600 individuals) and the binary nature of the phenotype (which reduces statistical power). However, this test confirms that the framework runs as expected, ingests ancestry-informed covariates, and returns results showing the statistical significance of the tested interaction, the number of PCs used (generated from the preprocessing step), and the phenotype distribution (mean and standard deviation).

2.3 Generation of imputation panels for combined sequencing with biobank data

Implementation

Building upon GLIMPSE2's tutorial (Olivier Delaneau, n.d.; Rubinacci et al., 2021, 2023), which provides bash script snippets for generating reference panels from a single chromosome (specifically, chromosome 22 of The 1000 Genomes Project b38 data from the EBI FTP site (Consortium, 2015)), we developed a scalable Nextflow pipeline capable of processing all chromosomes in our dataset. Due to the complexities associated with chromosome X, it was excluded from this project.

Dataset

The pipeline utilizes the combined, phased HGDP+1kGP haplotype dataset from the HGDP and the 1000 Genomes Project, available in the gnomAD public cloud folders that can be accessed from either Google Cloud Platform (gs://gcp-public-data–gnomad/resources/hgdp_1kg/phased_haplotypes_v2) or Amazon Web Services (s3://gnomad-public-us-east-1/resources/hgdp_1kg/phased_haplotypes_v2
). The dataset is made up of 4091 subjects from diverse ancestries.

The pipeline comprises four main steps:

1. Conversion of multiallelic sites: Transform all multiallelic sites into biallelic sites, retaining both single nucleotide polymorphisms (SNPs) and insertions/deletions (indels)

2. Extraction of site information: Extract site information for the entire cohort ignoring specific genotype calls

3. Chunking reference data: Divide the reference data using GLIMPSE2_chunk and prior mapping information provided in the GLIMPSE2 repository

4. Splitting reference chromosomes: Segment the reference chromosomes into binary chunks for all chromosomes

The software tools employed in this pipeline include bcftools (Danecek et al., 2021; Samtools, 2025), GLIMPSE2 (Olivier Delaneau, n.d.; Rubinacci et al., 2021, 2023), Nextflow (Di Tommaso et al., 2017; Tommaso et al., 2017), and Docker (Docker, 2025; Merkel, 2014).

Operation

Users can execute the Nextflow pipeline on their chosen phased, WGS dataset to create a customized reference panel. The pipeline's modular design facilitates straightforward integration into existing workflows, enhancing its adaptability for various research applications.

2.4 Rapid Longitudinal Analysis of Public Health Data

Implementation Details

Our causal discovery pipeline consists of three major steps, with each step contributes to the larger goal of automating causal inference from electronic health records (EHRs), ensuring efficient preprocessing, scalable computation, & meaningful insights into ICU patient outcomes.

1. Parsing User Input YAML File

2. Running Tetrad (Ramsey et al., 2018) on Preprocessed Data

3. Visualizing Tetrad Output

Step1: Parsing User Input YAML File

This step ensures that causal discovery is performed on a relevant, well-structured dataset, reducing noise and improving interpretability.

Input:

- A YAML configuration file that includes:
  - Patient Filtering Criteria (e.g., diagnosis codes, admission type)
  - Selected Clinical Features (e.g., lab results, vital signs, microbiological events)
  - Causal Discovery Parameters (e.g., algorithm selection, conditional independence test)
- An SQLite database that contains the MIMIC-III v1.4 CareVue subset dataset (A. Johnson et al., 2022), containing patient data spread across 27 relational tables.

Processing:

- Load the YAML file and extract the user-defined filters.

- Query the relevant MIMIC-III v1.4 tables (e.g., DIAGNOSES_ICD, LABEVENTS, MICROBIOLOGYEVENTS) to extract data matching the filtering criteria.
- Join multiple tables to create a cohort dataset, ensuring patient records are structured appropriately for causal analysis.
- Convert categorical variables into appropriate numerical representations (e.g., encoding comorbidities, ICU transfers).
- Store the preprocessed dataset in an SQL database for efficient querying and downstream analysis.

Output: A cleaned, preprocessed dataset filtered based on user criteria, ready for causal discovery & a SQL database containing structured, efficiently retrievable patient records.

Step2: Running Tetrad on Preprocessed Data

This step applies advanced statistical and machine learning techniques to uncover potential causal relationships in ICU patient data, moving beyond correlation-based analyses.

Input:

- The preprocessed dataset(s) from Step1
- User-defined causal discovery parameters (e.g., choice of algorithm, significance thresholds).

Processing:

- Convert the structured dataset into a format compatible with Tetrad, a causal discovery tool.
- Run the selected causal discovery algorithm (e.g., PC Algorithm, FCI, GES) on the dataset.
- Generate graphical causal models (DAGs) representing inferred causal relationships among variables.
- Store Tetrad's raw output, including adjacency matrices, edge confidence scores, and causal directionality data.

Output: Causal graphs (DAGs) illustrating the relationships between clinical variables & a tabular output listing causal dependencies, effect sizes, and statistical confidence measures.

Step3: Parsing Tetrad Output for Analysis

This step ensures that the discovered causal relationships are accessible, interpretable, and useful for clinical decision-making, enabling a data-driven approach to patient care.

Input:

- Tetrad output file(s) (causal graphs, adjacency matrices, tabular causal relationships) from Step2
- Original dataset for contextual mapping of identified causal relationships

Processing:

- Parse adjacency matrices to extract variable relationships and confidence scores.

- Convert the causal graph into a human-readable format (e.g., directed acyclic graph visualization).
- Aggregate findings to identify key drivers of patient outcomes (e.g., mortality, readmission risk).
- Generate summary statistics and reports for clinical interpretation.

Output: Graphical representations of causal relationships between clinical features along with tabular summaries highlighting statistically significant causal dependencies.

How to use it?

Open a bash window and execute the following commands to run the project: #!/bin/bash # Exit immediately if a command exits with a non-zero status set -e # Input to this step: User_input_yaml.txt echo "Step 1: Generating example user input..." Rscript GenerateExampleUserInput.r echo "Creating SQLite database..." bash Create_SQLite_DB.sh echo "Parsing user input and filtering dataset tables..." Rscript ParseUserInput.r # Input to this step (Output from previous step): Knowledge.txt echo "Step 2: Running Tetrad with the parsed input..." Rscript Run_tetrad_from_yaml.r echo "Step 3: Visualizing the Tetrad output..." Rscript s_visualize_tetrad_output_RS_AMP.R echo " Done!"

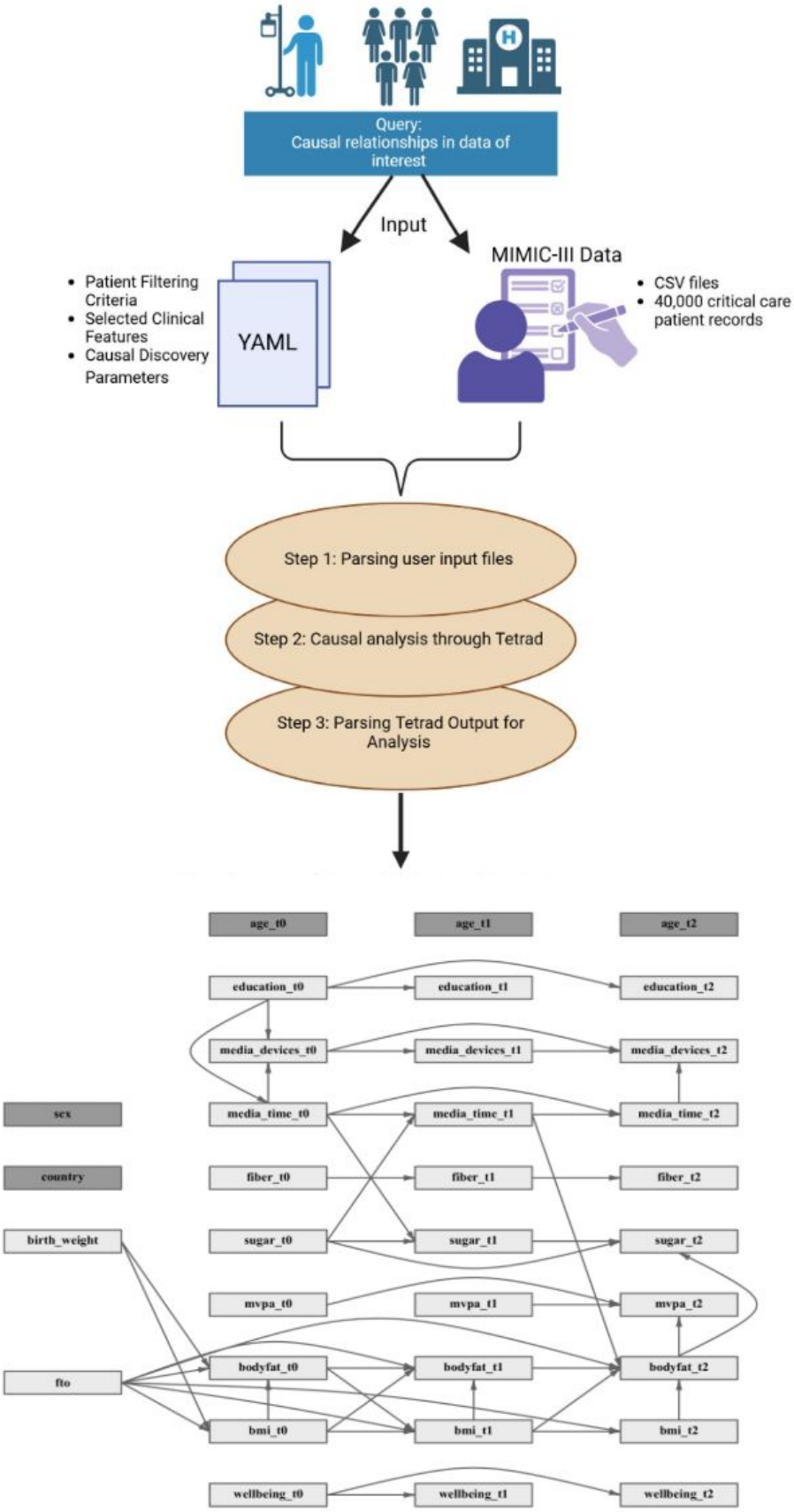

Figure 1: Overview of the methodology implemented

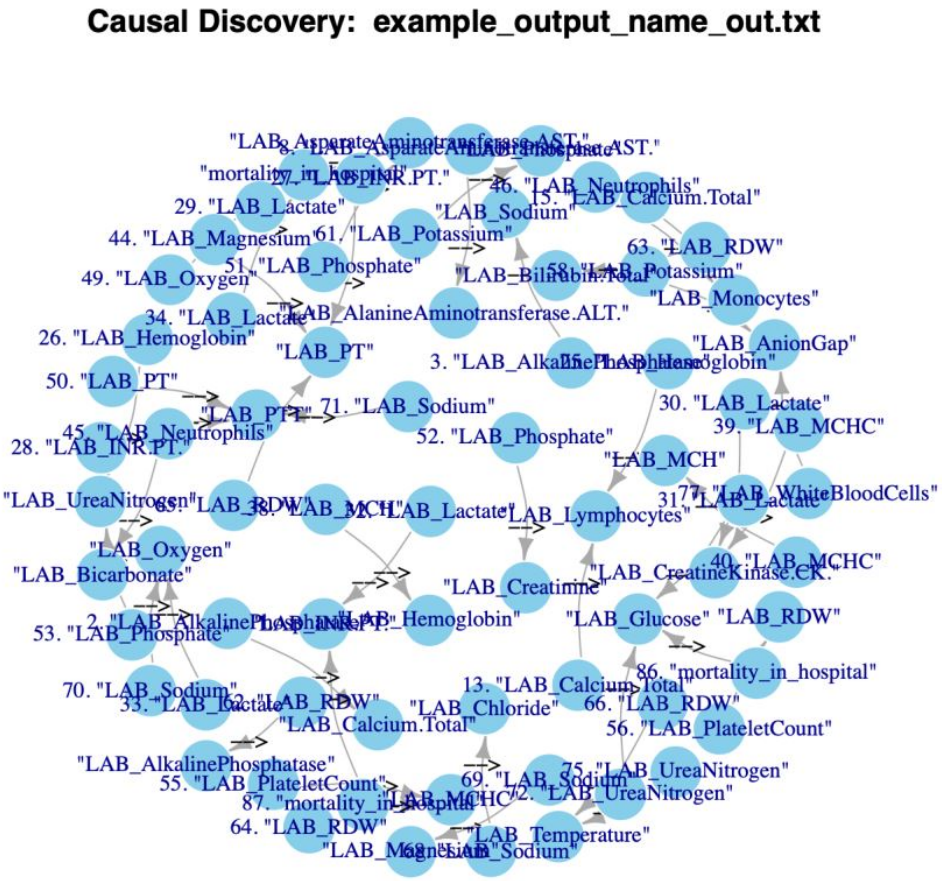

Figure 2: Plot of Tetrad-discovered causal relationships from MIMIC-III CareVue subset (A. Johnson et al., 2022)

2.5 Lenski-esque GNN Competition Trials

During the hackathon, we used protein-protein interaction data along with gene ontology, tissue-specific gene expression, and mutation rate from public databases: The Human Protein Atlas (Pontén et al., 2008), Genomic Data Commons (Heath et al., 2021) and the STRING database (Szklarczyk et al., 2021). In addition, we used disease-association data from the DisGeNET database (Piñero et al., 2020) as positive-label data for our training set.

We used the data preprocessing pipeline from geneGRAGNN, which includes the following steps:

1. *import_dgn.py* - provides gene-disease association scores and evidence index scores from DisGeNET

2. *import_gdc.py* - provide node features (mutation rates for a specific disease) from the National Institute of Health Genomic Data Commons data

3. *import_hpa.py* - provide node features (gene and RNA expression) from the Human Protein Atlas

4. *import_string.py* - provides protein-protein interactions from the STRING database

5. *create_node2vec_embeddings.py* - creates graph embeddings using node2vec

6. *main_data_pipeline.ipynb* - generates the final input file for geneDRAGNN

We trained a suite of networks of two types: the first was the original architecture of geneDRAGNN (Awni Altabaa et al., 2025); the second was the original architecture of geneDRAGNN with additional layers implemented in the PyTorch Geometric library [@[Fey_Fast_Graph_Representation_2019], such as:

- conv.GCNConv - the simplest GNN layer
- conv.SAGEConv - leverages node feature information (e.g., text attributes) to efficiently generate node embeddings for previously unseen data
- conv.GATConv - enables (implicitly) specifying different weights to different nodes in a neighborhood, by stacking layers in which nodes are able to attend over their neighborhoods' features,
- conv.TransformerConv - incorporates feature and label propagation at both training and inference time
- conv.HGTConv - designed for heterogeneous graphs

Following the methods and parameters outlined in geneGRAGNN, we extracted features from our input data using node2vec and created a graph network with the positive-unlabeled nodes [shi2021maskedlabelpredictionunified].

Methods – Operation (how do people use it?)

For scripts to install geneDRAGNN, set up the environment and run the experiments, see https://github.com/collaborativebioinformatics/LenskAI. We illustrate the modifications to the geneDRAGNN code in the README file. We provide the *models_hackathon.py* and *train_gnn_model_hackathon.py* scripts that we used for the experiments.

Methods – Extension (how can people extend it?)

Everyone is welcome to clone the repository and build upon the project, for example, to extend the experiments of evolving networks to include different combinations of layers. Furthermore, it is worth examining how the choice of the input data impacts the model performance. In other words, we hypothesize that neural networks can benefit from additional input data, however, it is often unclear what information is actually extracted from the data. Therefore, this project might additionally include training the neural network with several biological datasets of varying size and biological information, followed by using an explainable AI (XAI) method. Such XAI methods compute explanation subgraphs that show the impact of network patterns and node features on the output. With explanation subgraphs, it might be possible to examine each experiment more closely, hence explaining what biological information is leveraged by the neural network while making predictions.

2.6 Population-Specific Multiomics Graph Analysis of ACE Protein Expression

Methods – Implementation (how did you build it?)

This workflow aims to generate population-specific genome graphs that highlight the genetic variants influencing the expression of a target protein. These graphs are constructed based on pQTL (protein quantitative trait loci) data and genomic annotations, ultimately representing how different genetic variations impact protein expression across populations. We systematically analyze chromosome-specific variants affecting ACE protein expression using protein quantitative trait loci (pQTL) data (B. B. Sun et al., 2023). We integrate this data with the latest human

reference genome (GRCh38.p14) (GENCODE, 2024) and functional genomic annotations from GTF files to pinpoint coding and regulatory variants . To visualize population-specific genetic architectures, we construct disconnected genome graphs, where nodes represent genetic variants with key attributes such as effect size (Beta), p-value, and functional annotations. By comparing these graphs across populations, we aim to uncover distinct genetic influences on ACE protein expression, providing insights into population-specific regulatory mechanisms and their implications in precision medicine. This framework serves as a scalable approach for multi-omics graph analysis in complex trait studies. We build the workflow as follows:

Target Protein pQTL Data Processing The workflow begins with the processing of protein quantitative trait loci (pQTL) data for the target protein. This dataset consists of chromosome-specific variant files, one for each of the 23 chromosomes. These files are used to identify genetic variants associated with the target protein. The variants are extracted and organized by chromosome to facilitate downstream filtering and analysis. This step ensures that all relevant genetic information for the target protein is included in the pipeline.

Filtering Variants Based on User Input The extracted pQTL data is filtered based on user-defined criteria, which currently involves parsing the INFO column of the pQTL dataset. The filtering criteria can be customized to include specific variant properties, such as allele frequency, effect size, or functional annotations. This step reduces the dataset to variants that meet user-specified thresholds, ensuring that only biologically or clinically relevant variants are included in subsequent analyses.

Gene Annotations and Variant Mapping Gene annotation data in GTF (General Transfer Format) is used to map filtered variants to their corresponding genomic features. For each variant, its position is checked against annotated regions such as genes, untranslated regions (UTRs), and exons. If a variant falls within one of these regions, additional information such as the Ensembl gene ID is retrieved and linked to the variant. The current implementation focuses on protein-coding genes, but this can be extended to other gene types based on user input. This step ensures accurate mapping of variants to functional genomic elements. We use the Entrez module in BioPython for this step. (Cock et al., 2009; Contributors, 2000--2025, 2007--2025)

Gene Sequence Extraction The reference sequence for each gene identified in the previous step is retrieved using its Ensembl gene ID. This sequence forms the basis for constructing a linear "gene graph." The gene graph represents the reference sequence as a series of connected nodes, where each node corresponds to a segment of the sequence. This step provides a foundational structure for integrating variant information into the graph.

Variant Integration into Gene Graphs Variants mapped to genes are integrated into their respective gene graphs by modifying or adding nodes and edges. Each node in the graph stores metadata such as nucleotide sequence, length, position, strand orientation, chromosome number, and genomic feature type (e.g., gene, exon, UTR). For variant nodes, additional properties such as base effect (e.g., SNPs, insertions, deletions) and statistical significance (e.g., logP value) are also stored. Reference sequence nodes do not include these additional properties. All variants are connected to both upstream and downstream nodes in the graph to ensure continuity.

Construction of Target Protein Genome Graph The final step involves combining all gene graphs associated with the target protein into a comprehensive genome graph. This graph incorporates both reference sequences and variant information for all relevant genes across chromosomes. The resulting structure provides a holistic view of genetic variation affecting the target protein and enables downstream analyses such as path traversal or visualization of alternative haplotypes.

Parallelization To optimize performance, especially when processing large datasets or multiple chromosomes simultaneously, parallelization is explored within this workflow. Computationally intensive steps such as filtering variants or constructing gene graphs can be parallelized

across multiple processors or distributed computing environments. This ensures scalability and efficiency when handling high-throughput sequencing data. We use PyTorch in some elements of the pipeline which gives GPU parallelization. (PyTorch contributors, 2016--2025).

Methods – Operation (how do people use it?)

Refer to the GitHub repository (https://github.com/collaborativebioinformatics/Multiomic_graph) for instructions on data processing. This will generate a .tsv file that will be used in the graph generation step.

Once the filtered tab separated value (.tsv) file is created, utilize the graph.py executable script to generate your graphs. Provide the script with the variant .tsv file, along with an email to use for NCBI API calls. Outputs will be stored in an "outputs" folder which contains a compressed PyTorch data file, and a text file indicating the sequence of the gene graphs. To open the compressed file, utilize gzip and make sure to set "weights_only=False" when calling torch.load().

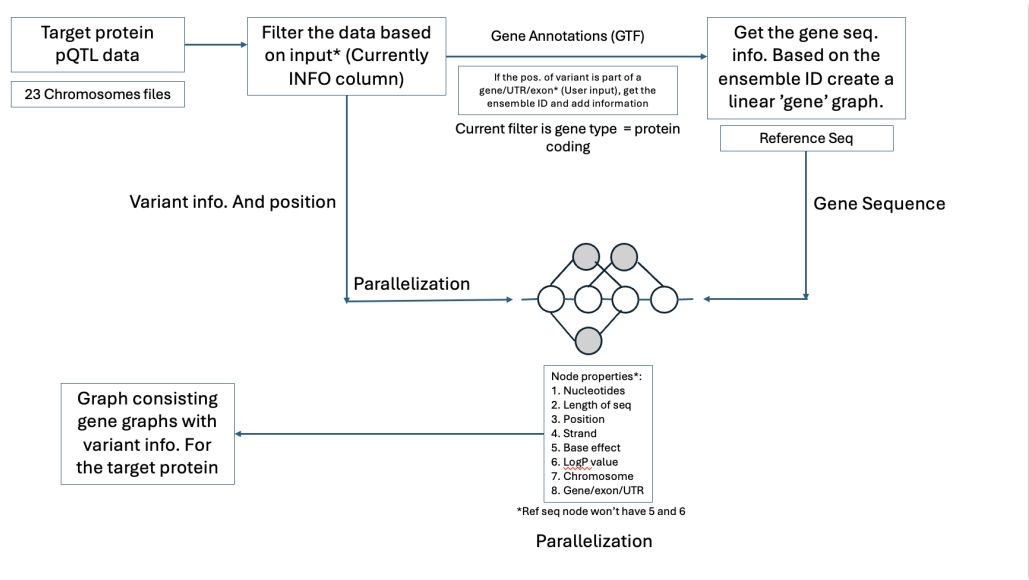

Figure 3: Data integration and graph construction

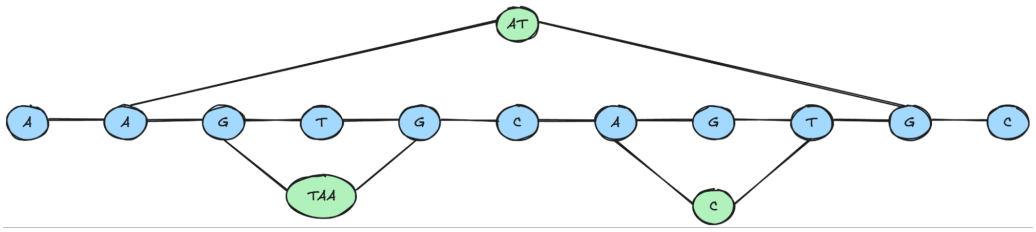

Figure 4: Genome graph

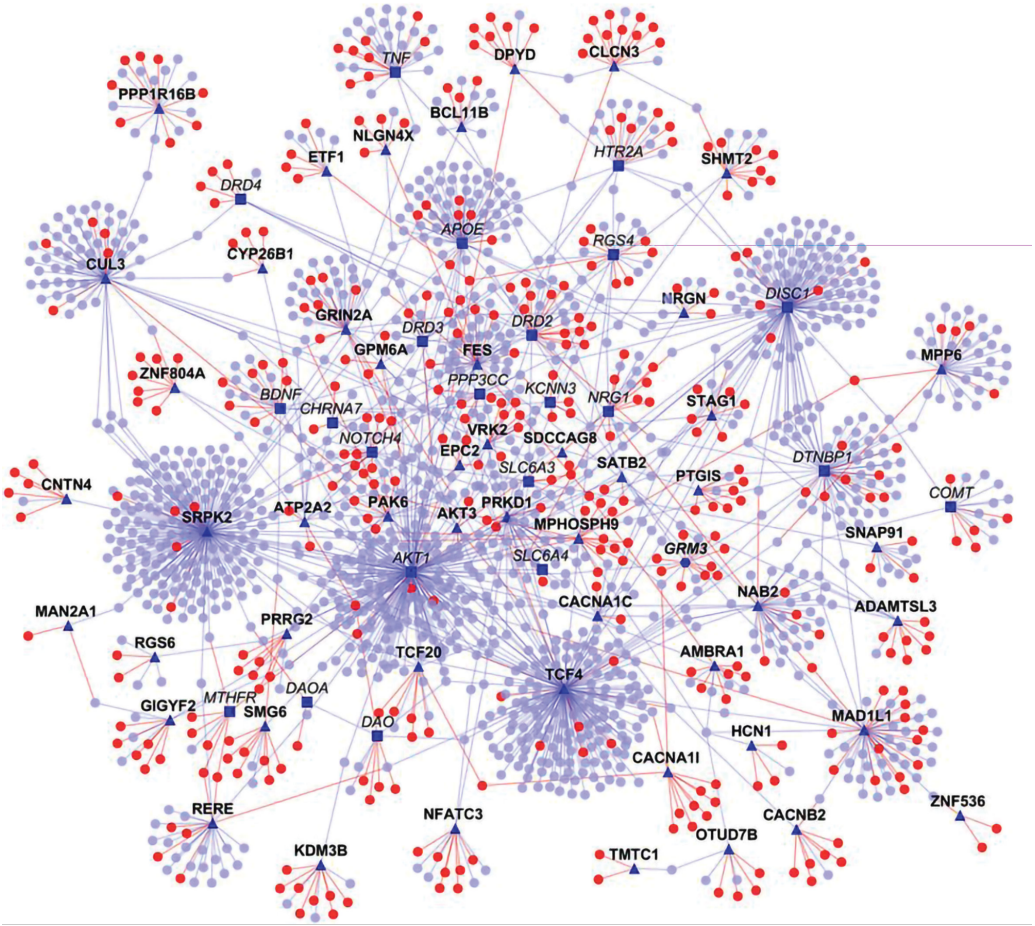

Figure 5: Protein network (from (LeMieux, 2025))

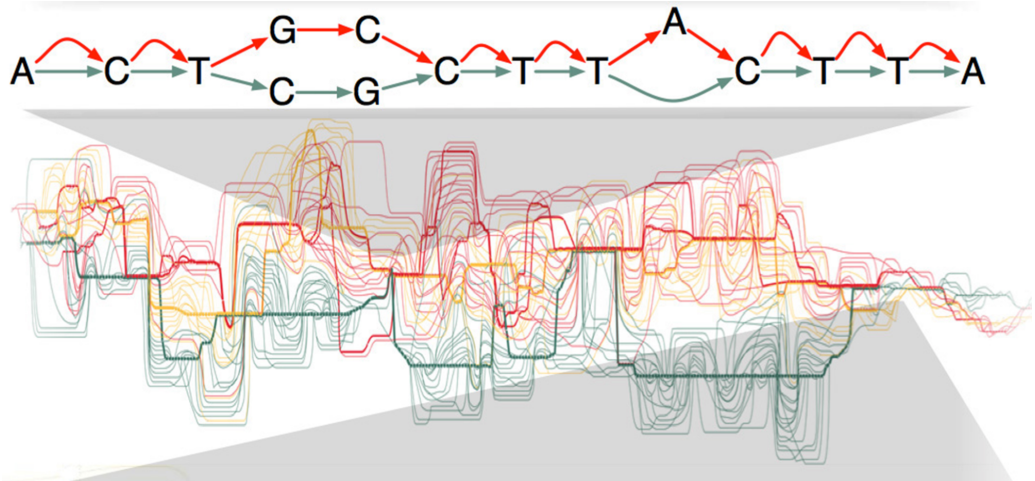

Figure 6: Gene variants in population (from (Zimmer, 2016))

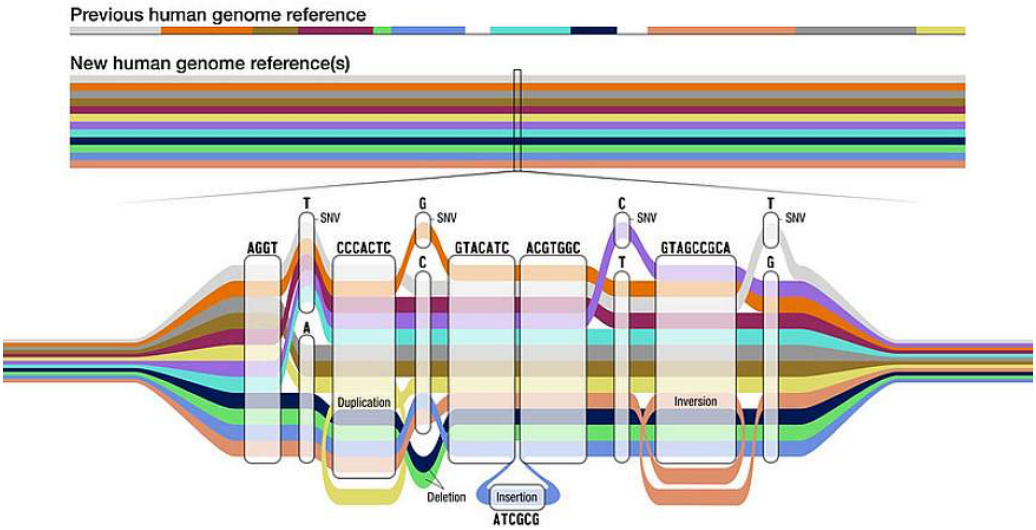

Figure 7: Population gene graphs (from (National Institutes of Health, 2023))

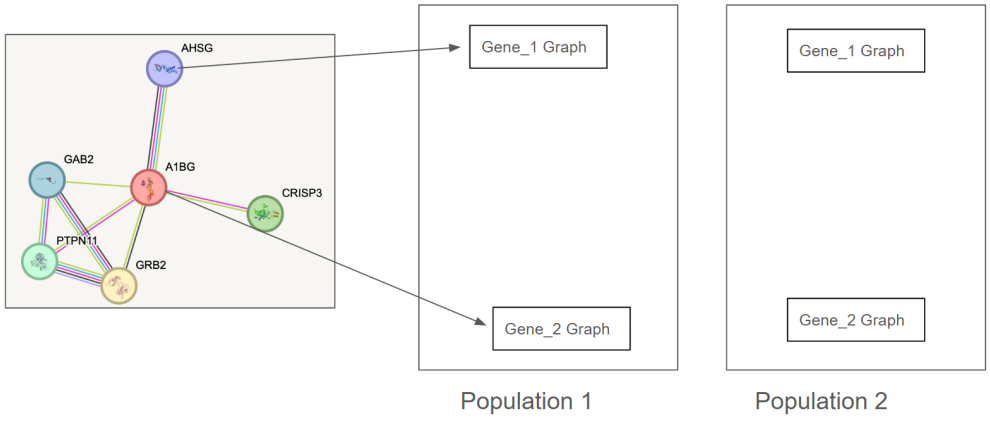

Figure 8: Generate gene-specific graphs for variant annotation and analysis

3. Discussion and Future Directions

3.1 Clustering of haplotype matrices

Although we were able to establish methods to perform clustering and annotation analysis of example ARG-Needle data, a sticking point in our overall pipeline development was the difficulty of successfully producing ARGN files from the chromosome-level The 1000 Genomes Project (Consortium, 2015).

One potential cause of the problems we've been experiencing with ARG-Needle in this project could be because we've been trying to work with a highly diverse population from The 1000 Genomes Project. ARG construction with ARG-Needle relies on threading of samples into the graph based on similarity to other individuals, but ARG-Needle was primarily tested on samples from the UK Biobank, representing a highly homogenous population. For our purposes, selecting a more homogenous subpopulation, such as a British European or Dai Chinese subpopulation, from The 1000 Genomes Project data on which to run ARG construction could

work better. Additional potential approaches to this issue could include using alternative tools like GenoTools or generating brute-force similarity matrices from haplotype data.

While we also examined clusters of shared variants in two genes known to be highly variable (beta-defensin and HLA-A), future directions should further expand this analysis to more known highly variable genes, and should also consider including linkage disequilibrium analysis. Such analysis could identify linked blocks of variants useful for haplotype determination. Identifying linkage blocks based on different adjacent variant linkage thresholds, in addition to consideration of linkage block diversity amongst population genomic data, may provide data useful for downstream machine learning identification of haplotype clusters corresponding to particular subpopulations.

3.2 Cis and trans effects of haplotypes on rare variants penetrance with StructLMM adapted for Gene-Gene interaction analysis

Detecting GxG interactions remains one of the more elusive goals in statistical genetics. While single-variant models have matured significantly through GWAS, extending these frameworks to capture the combinatorial effects of interacting loci has proven difficult, often due to confounding and subtle effect sizes. In this project, we adapted the StructLMM framework, originally developed for structured gene-environment (GxE) interaction analysis, to explore GxG interactions, using local ancestry principal components (PCs) as structured proxies for haplotypic variation.

This choice is motivated by the biological observation that genetic interactions can occur in cis (nearby variants within the same regulatory or linkage context) and trans (distal interactions across loci or chromosomes). By extracting ancestry PCs from a genomic region of interest, we implicitly summarize ancestry and local haplotypic structure to capture the combined influence of linked variants that may modulate a focal SNV. In essence, these PCs allow us to test whether the penetrance of a specific variant is modified by its surrounding genetic environment.

Our application of the method to a synthetic dataset from HAPNEST (Wharrie et al., 2022) served as a proof-of-concept. While the result ($p = 0.86$) did not suggest any statistically significant interaction between the query SNV and the regional ancestry PCs, this is not a failure of the model per se.

The value of this demonstrated framework lies in its scalability and flexibility. It provides a path to exploring GxG effects, particularly in cohorts with admixed ancestries or rich regional genomic structures. With future refinements, the method can serve as a practical tool for uncovering complex genetic interactions in human traits. Adapting the method and validating its application to large-scale biobanks, such as the UK Biobank (https://www.ukbiobank.ac.uk/) or All of Us Research Program (https://allofus.nih.gov/) or disease-specific studies will determine its real-world performance.

Several avenues exist for extending and refining this work. One avenue is to scale the analysis to larger sample sizes, ideally using the full HAPNEST dataset or other real-world biobank-scale cohorts with diverse ancestry backgrounds. Increased statistical power would make it feasible to detect subtler interaction effects and evaluate whether the framework can recover known or simulated GxG interactions.

Second, the current implementation treats local ancestry PCs as an abstract structured covariate. Future work could compare this to explicit haplotype-based models, such as those incorporating phased haplotypes, to assess whether PCs effectively approximate the local genetic background or lose signal due to dimensionality reduction.

Third, expanding the model to accommodate categorical or multi-class phenotypes and testing robustness across different phenotype types (e.g., quantitative vs. binary) could further generalize the tool's applicability. In this initial test, only one phenotype was used; running across all nine HAPNEST traits could reveal more favorable settings for detecting interactions.

From a software perspective, it would also be valuable to package the code into a reproducible module or wrapper to allow users to plug in query variants and regions without reworking preprocessing steps. As part of this, the selection of PCs could be automated rather than separated.

3.3 Generation of imputation panels for combined sequencing with biobank data

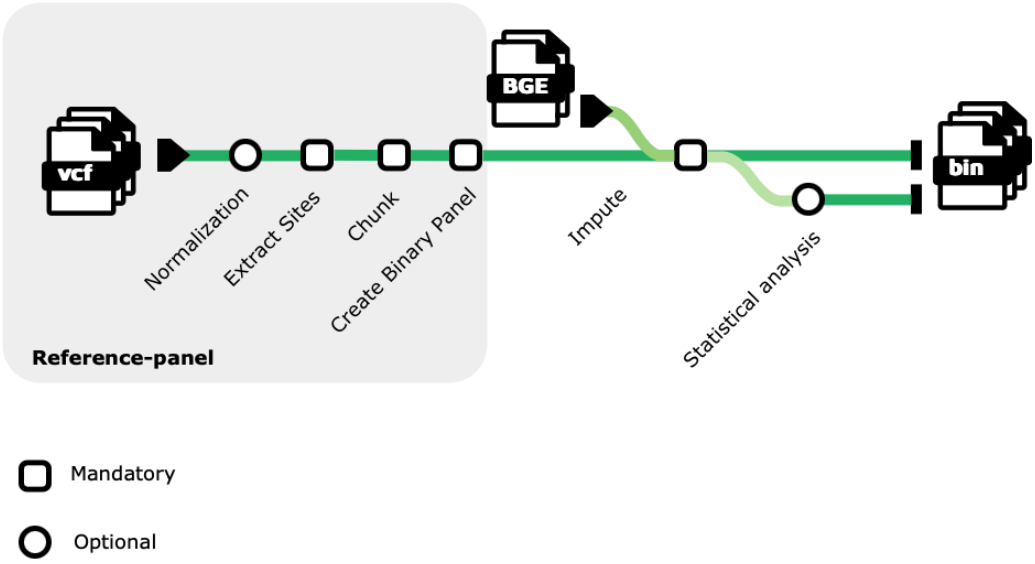

Figure 9: Nextflow pipeline

The workflow has been successfully tested on local systems, on-premises HPC systems, DNANexus, and Google Cloud Platform (GCP). On DNANexus, the workflow takes <1 hour to create the final binary reference files for our example dataset using 22 concurrent operators on 4 core virtual machines (mem2_ssd1_v2_x4). The compute cost of generating the reference panel was USD $1.50.

Future directions include inclusion of chrX into the workflow, adding additional platforms (AWS, Azure), and providing the ability to work on additional datasets. These advancements would further augment the pipeline's robustness and utility in genomic research. As Biobank scale datasets continue to grow, (UK Biobank, All of Us, Mexico City Biobank, Our Future Health, FinnGen, Japan Biobank, and others), the feasibility of merging the raw data between these resources may be low, but each could potentially provide imputations panels for public use. This workflow could facilitate the generation of those panels.

3.4 Rapid Longitudinal Analysis of Public Health Data

Using the MIMIC-III v1.4 dataset (Computational Physiology, 2016; Goldberger et al., 2000; Alistair Johnson et al., 2016; A. E. W. Johnson et al., 2016), 87 causal relationships were found. The relations along with their interpretations are mentioned in the appendix. The primary outcome of our methodology was a significant reduction in Tetrad computation time (Ramsey et al., 2018). By leveraging user input, the pipeline efficiently filters large datasets, extracting only the necessary rows and columns required for analysis. This preprocessing step drastically reduces the data size before running Tetrad, optimizing computational efficiency. Furthermore, post-filtering, Tetrad is executed in parallel across multiple data chunks, allowing for faster processing and better scalability. The pipeline utilizes multiple features and clinical outcomes from Electronic Medical Records (EMR) to intelligently guide the execution of Tetrad,

ensuring that causal inference is both data-driven and contextually relevant. The methodology significantly reduces the computational burden of running Tetrad by filtering the dataset based on user-defined criteria. This ensures that only relevant subsets of data are processed, making causal inference more efficient.

By splitting the dataset into manageable chunks and executing Tetrad in parallel, the methodology leverages efficient processing, reducing execution time and improving scalability for large datasets.

The use of Electronic Medical Records (EMR) features and outcomes guides data selection, ensuring that the input data remains clinically meaningful while eliminating irrelevant noise.

The approach is highly adaptable, allowing it to be extended to other large-scale datasets, including biological data (e.g., SNPs, gene expression), without major modifications. Future improvements can integrate dimensionality reduction techniques like PCA, which could further compress large datasets while preserving essential patterns, thus improving computational efficiency.

This pipeline has significant potential for expansion and adaptation to newer challenges and datasets. Some key areas for future development include:

- The methodology could be extended to analyze biological datasets, such as single nucleotide polymorphisms (SNPs) and gene expression data. By incorporating statistical models and causal inference techniques, we could explore genetic associations and uncover meaningful biological relationships.

- Implementing Principal Component Analysis (PCA) / Uniform Manifold Approximation and Projection (UMAP) or other feature selection methods could help optimize performance by reducing the dataset size while preserving key information (This can be taken from the output of other team). Automated hyperparameter tuning and scalable database queries could further enhance efficiency.

- The pipeline can be modified to handle different domains, such as electronic health records (EHRs), population health studies, or clinical trial data. Extending support for multi-modal datasets (e.g., combining text, images, and structured data) would make the pipeline more versatile.

- Adding predictive modeling techniques (e.g., random forests, deep learning, or Bayesian inference) could enhance its ability to identify patterns and forecast outcomes. Time-series modeling could be integrated for longitudinal health data analysis.

- Developing a graphical user interface (GUI) or API would make the pipeline accessible to researchers with minimal coding expertise. Integration with R Shiny or Jupyter Notebooks could enable interactive exploration of results.

3.5 Lenski-esque GNN Competition Trials

Large biological networks such as protein-protein interaction networks or disease-gene association networks provide essential information about relationships and interactions between biomolecules. With the growing amount of such data, new bioinformatics approaches are needed. We conducted a project at the CMU / DNAnexus Hackathon 2025 to examine how we can leverage graph neural networks to extract biological insights from network-type of data. During the hackathon we aimed at training and "evolving" a graph neural network geneDRAGGN for disease gene prioritization using public protein-protein interaction data (STRING database) and disease-gene association data (DisGeNET database). Our results suggest that imputing such large biological networks into graph neural networks is challenging due to computational requirements of such algorithms despite using an efficient cloud

infrastructure. Our preliminary results present a unique set of top 10 genes that predict worse disease outcomes in lung cancer patients.

Looking forward, we plan to explore possible solutions to the challenge of imputing large biological networks into graph neural networks. In addition, we emphasize the importance of developing algorithms for reducing the size of network files so that the computation is more memory-efficient, yet no significant biological information is neglected or removed.

3.6 Population-Specific Multiomics Graph Analysis of ACE Protein Expression

This workflow aims to generate population-specific genome graphs that highlight the genetic variants influencing the expression of a target protein. These graphs are constructed based on pQTL (protein quantitative trait loci) data and genomic annotations, ultimately representing how different genetic variations impact protein expression across populations.

Advancements in multi-omics analysis have significantly enhanced our ability to investigate the genetic and molecular mechanisms underlying complex traits and diseases ((Hasin et al., 2017); (Misra et al., 2019)). Protein quantitative trait loci (pQTL) studies play a crucial role in linking genetic variation to protein expression, providing key insights into gene regulation at the protein level (Suhre et al., 2011; Benjamin B. Sun et al., 2018). However, traditional variant mapping approaches often fail to account for population-specific genetic architectures and the broader functional context of these variations (Garrison et al., 2018; Wojcik et al., 2019).

Previous studies have demonstrated that pQTLs can influence protein expression through multiple mechanisms, including transcriptional regulation, mRNA stability, and post-translational modifications (Chick et al., 2016; Suhre et al., 2011). However, mapping these associations accurately remains challenging due to reference genome biases, the presence of highly polymorphic regions, and population-level genetic diversity (Benjamin B. Sun et al., 2018). While graph-based genome representations have been proposed to mitigate mapping errors and improve variant calling in diverse populations (Eizenga et al., 2020; Garrison et al., 2018), there is still a critical need for scalable multi-omics approaches that integrate pQTL data with functional genomic annotations to identify population-specific regulatory networks.

Here, we present a multi-omics graph-based framework for studying ACE protein expression across Asian and African populations. Our approach integrates pQTL data, the latest human reference genome (GRCh38.p14), and functional genomic annotations (GTF files) to construct population-specific genome graphs. Each subgraph represents gene-level regulatory interactions, incorporating chromosome-specific variant effects, beta coefficients, p-values, and functional annotations. By leveraging graph-based modeling, we aim to uncover both shared and population-specific influences on ACE protein regulation, ultimately providing novel insights into genetic variation and its impact on protein expression.

This framework not only enhances the accuracy of variant interpretation but also establishes a scalable method for multi-omics pathway analysis, facilitating discoveries in precision medicine and systems biology.

Future work will focus on parallelizing graph generation to distribute construction across multiple genes, thereby accelerating processing for large datasets. We plan to implement advanced filtering options to allow selection by gene lists, features, gene types, and genomic coordinates to produce problem-specific graphs. Additionally, integrating Graph Neural Networks (GNNs) for variant annotation and downstream analysis will be pursued, along with optimizing computational efficiency to ensure scalability for large-scale genomic studies.

4. Data and Software Availability

All code and required software stacks are provided in the following GitHub repositories, which may include additional links to data repositories and Jupyter Notebooks.

If you or your colleagues are interested in collaborating on these or similar projects in a hackathon or professional setting, please contact ben.busby@gmail.com. If you have technical questions or issues, please put an issue into one of the GitHub repositories listed below.

4.1 Clustering of haplotype matrices: https://github.com/collaborativebioinformatics/Haplotype_matrix_clustering

4.2 Cis and trans effects of haplotypes on rare variants penetrance with StructLMM adapted for Gene-Gene interaction analysis: https://github.com/collaborativebioinformatics/Cis_and_trans_effects_on_variant_penetrance

4.3 Generation of imputation panels for combined sequencing with biobank data: https://github.com/collaborativebioinformatics/Blended_seq_imputation

4.4 Rapid Longitudinal Analysis of Public Health Data: https://github.com/collaborativebioinformatics/Longitudinal_emr_accleRation

4.5 Lenski-esque GNN Competition Trials: https://github.com/collaborativebioinformatics/LenskAI

4.6 Population-Specific Multiomics Graph Analysis of ACE Protein Expression: https://github.com/collaborativebioinformatics/Multiomic_graph

5. Acknowledgements

We would like to thank Carnegie Mellon University Libraries and DNAnexus Inc. for organizing and hosting this event. DNAnexus Inc. provided cloud computing resources, as well as support from Theresa Wohlever. Tom Hughes provided logistical support.

Grant Funding JK is supported by the France 2030 state funding managed by the National Research Agency with the reference "ANR-22-PEPRSN-0013".

6. Appendix

Rapid Longitudinal Analysis of Public Health Data

87 causal relationships found are as follows:

Graph Edges:
1. "LAB_AlkalinePhosphatase" o-> "LAB_Bilirubin.Total"
2. "LAB_AlkalinePhosphatase" --> "LAB_Calcium.Total"
3. "LAB_AlkalinePhosphatase" --> "LAB_Sodium"
4. "LAB_AnionGap" <-> "LAB_Bicarbonate"
5. "LAB_AnionGap" <-> "LAB_Creatinine"
6. "LAB_AnionGap" <-> "LAB_Lactate"
7. "LAB_AnionGap" <-> "LAB_Phosphate"
8. "LAB_AsparateAminotransferase.AST." --> "LAB_AlanineAminotransferase.ALT."
9. "LAB_Basophils" o-> "LAB_Eosinophils"
10. "LAB_Bicarbonate" <-> "LAB_Chloride"
11. "LAB_Bicarbonate" <-> "mortality_in_hospital"
12. "LAB_Calcium.Total" <-> "LAB_CreatineKinase.CK."
13. "LAB_Calcium.Total" --> "LAB_Lymphocytes"
14. "LAB_Calcium.Total" <-> "LAB_Magnesium"

15. “LAB_Calcium.Total” --> “LAB_Monocytes”
16. “LAB_CalculatedTotalCO2” o-> “LAB_BaseExcess”
17. “LAB_CalculatedTotalCO2” o-> “LAB_Bicarbonate”
18. “LAB_CalculatedTotalCO2” o-> “LAB_pCO2”
19. “LAB_Creatinine” <-> “LAB_UreaNitrogen”
20. “LAB_Eosinophils” <-> “LAB_Neutrophils”
21. “LAB_Eosinophils” <-> “LAB_PlateletCount”
22. “LAB_Eosinophils” <-> “mortality_in_hospital”
23. “LAB_Hematocrit” o-> “LAB_Hemoglobin”
24. “LAB_Hematocrit” o-> “LAB_Phosphate”
25. “LAB_Hemoglobin” --> “LAB_Lymphocytes”
26. “LAB_Hemoglobin” --> “LAB_UreaNitrogen”
27. “LAB_INR.PT.” --> “LAB_PT”
28. “LAB_INR.PT.” --> “LAB_PTT”
29. “LAB_Lactate” --> “LAB_AsparateAminotransferase.AST.”
30. “LAB_Lactate” --> “LAB_CreatineKinase.CK.”
31. “LAB_Lactate” --> “LAB_Glucose”
32. “LAB_Lactate” --> “LAB_INR.PT.”
33. “LAB_Lactate” --> “LAB_Oxygen”
34. “LAB_Lactate” --> “mortality_in_hospital”
35. “LAB_Lymphocytes” <-> “LAB_Monocytes”
36. “LAB_Lymphocytes” <-> “LAB_Neutrophils”
37. “LAB_Lymphocytes” <-> “mortality_in_hospital”
38. “LAB_MCH” --> “LAB_Hemoglobin”
39. “LAB_MCHC” --> “LAB_CreatineKinase.CK.”
40. “LAB_MCHC” --> “LAB_MCH”
41. “LAB_MCV” o-> “LAB_MCH”
42. “LAB_MCV” o-> “LAB_Potassium”
43. “LAB_MCV” o-o “LAB_pO2”
44. “LAB_Magnesium” --> “LAB_PT”
45. “LAB_Neutrophils” --> “LAB_Bicarbonate”
46. “LAB_Neutrophils” --> “LAB_Monocytes”
47. “LAB_Neutrophils” <-> “LAB_PlateletCount”
48. “LAB_Neutrophils” <-> “LAB_WhiteBloodCells”
49. “LAB_Oxygen” --> “mortality_in_hospital”
50. “LAB_PT” --> “LAB_PTT”
51. “LAB_Phosphate” --> “LAB_AsparateAminotransferase.AST.”
52. “LAB_Phosphate” --> “LAB_Creatinine”
53. “LAB_Phosphate” --> “LAB_UreaNitrogen”
54. “LAB_PlateletCount” <-> “LAB_Lactate”
55. “LAB_PlateletCount” --> “LAB_MCHC”
56. “LAB_PlateletCount” --> “LAB_RDW”
57. “LAB_PlateletCount” <-> “mortality_in_hospital”
58. “LAB_Potassium” --> “LAB_AnionGap”
59. “LAB_Potassium” <-> “LAB_MCHC”
60. “LAB_Potassium” <-> “LAB_Magnesium”
61. “LAB_Potassium” --> “LAB_Phosphate”
62. “LAB_RDW” --> “LAB_AlkalinePhosphatase”
63. “LAB_RDW” --> “LAB_Bilirubin.Total”
64. “LAB_RDW” --> “LAB_MCHC”
65. “LAB_RDW” --> “LAB_PT”
66. “LAB_RDW” --> “LAB_Temperature”
67. “LAB_RedBloodCells” o-o “LAB_Hematocrit”
68. “LAB_Sodium” --> “LAB_Chloride”
69. “LAB_Sodium” --> “LAB_Magnesium”

70. “LAB_Sodium” --> “LAB_Oxygen”
71. “LAB_Sodium” --> “LAB_PTT”
72. “LAB_UreaNitrogen” --> “LAB_Glucose”
73. “LAB_UreaNitrogen” <-> “LAB_Magnesium”
74. “LAB_UreaNitrogen” <-> “LAB_PlateletCount”
75. “LAB_UreaNitrogen” --> “LAB_Temperature”
76. “LAB_UreaNitrogen” <-> “mortality_in_hospital”
77. “LAB_WhiteBloodCells” --> “LAB_AnionGap”
78. “LAB_WhiteBloodCells” o-> “LAB_PlateletCount”
79. “LAB_WhiteBloodCells” <-> “mortality_in_hospital”
80. “LAB_pH” o-> “LAB_BaseExcess”
81. “LAB_pH” o-o “LAB_CalculatedTotalCO2”
82. “LAB_pH” o-> “LAB_MCHC”
83. “LAB_pH” o-> “LAB_pCO2”
84. “LAB_pO2” o-> “LAB_BaseExcess”
85. “LAB_pO2” o-> “LAB_RDW”
86. “mortality_in_hospital” --> “LAB_Glucose”
87. “mortality_in_hospital” --> “LAB_INR.PT.”

Interpretation of results:
A --> B
present
A is a cause of B. It may be a direct or indirect cause that may include other measured variables. Also, there may be an unmeasured confounder of A and B.
absent
B is not a cause of A.
A <-> B
present
There is an unmeasured variable (call it L) that is a cause of A and B. There may be measured variables along the causal pathway from L to A or from L to B.
absent
A is not a cause of B. B is not a cause of A.
A o-> B
present
Either A is a cause of B, or there is an unmeasured variable that is a cause of A and B, or both.
absent
B is not a cause of A.
A o-o B
Exactly one of the following holds: (a) A is a cause of B, or (b) B is a cause of A, or (c) there is an unmeasured variable that is a cause of A and B, or (d) both a and c, or (e) both b and c.

References

Alamin, M., Sultana, M. H., Lou, X., Jin, W., & Xu, H. (2022). Dissecting complex traits using omics data: A review on the linear mixed models and their application in GWAS. *Plants*, *11*(23), 3277. https://doi.org/10.3390/plants11233277

Alharbi, F., Vakanski, A., Zhang, B., Elbashir, M. K., & Mohammed, M. (2025). Comparative analysis of multi-omics integration using graph neural networks for cancer classification. *IEEE Access*, *13*, 37724–37736. https://doi.org/10.1109/ACCESS.2025.3540769

Altabaa, A., Huang, D., Byles-Ho, C., Khatib, H., Sosa, F., & Hu, T. (2022). geneDRAGNN: Gene disease prioritization using graph neural networks. *2022 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 1–10. https://doi.org/10.1109/CIBCB55180.2022.9863043

Altabaa, Awni, Huang, D., Byles-Ho, C., Khatib, H., Sosa-Miranda, F., & Hu, T. (2025). *geneDRAGNN: Gene disease prioritization using graph neural networks*. https://github.com/geneDRAGNN/geneDRAGNN/blob/main/data/Readme.md

Barabási, A.-L., Gulbahce, N., & Loscalzo, J. (2011). Network medicine: A network-based approach to human disease. *Nature Reviews Genetics*, *12*(1), 56–68. https://doi.org/10.1038/nrg2918

Cavalli-Sforza, L. L. (2005). The human genome diversity project: Past, present and future. *Nature Reviews Genetics*, *6*(4), 333–340. https://doi.org/10.1038/nrg1596

Chang, C. C., Chow, C. C., Tellier, L. C. A. M., Vattikuti, S., Purcell, S. M., & Lee, J. J. (2015). Second-generation PLINK: Rising to the challenge of larger and richer datasets. *GigaScience*, *4*. https://doi.org/10.1186/s13742-015-0047-8

Chick, J. M., Munger, S. C., Simecek, P., Huttlin, E. L., Choi, K., Gatti, D. M., Raghupathy, N., Svenson, K. L., Churchill, G. A., & Gygi, S. P. (2016). Defining the consequences of genetic variation on a proteome-wide scale. *Nature*, *534*(7608), 500–505.

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., & Hoon, M. J. L. de. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163

Computational Physiology, M. L. for. (2016). *MIMIC-III documentation*. https://mimic.mit.edu/docs/about/

Consortium, T. 1000. G. P. (2015). A global reference for human genetic variation. *Nature*, *526*, 68–74. https://doi.org/10.1038/nature15393

Contributors, B. (2000--2025). *Biopython*. https://biopython.org.

Contributors, B. (2007--2025). *Bio.entrez package — biopython 1.76 documentation*. https://biopython.org/docs/1.76/api/Bio.Entrez.html.

Cordell, H. (2009). Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics*, *10*, 392–404. https://doi.org/10.1038/nrg2579

CRG, B. (2021). *General transfer format*. https://biocorecrg.github.io/PhD_course_genomics_format_2021/gtf_format.html

Danecek, P.others. (2021). Twelve years of samtools and bcftools. *GigaScience*, *10*(2), giab008. https://doi.org/10.1093/gigascience/giab008

DeFelice, M.others. (2024). *Blended genome exome (bge) as a cost efficient alternative to deep whole genomes or arrays*. https://doi.org/10.1101/2024.04.03.587209

Di Tommaso, P.others. (2017). Nextflow enables reproducible computational workflows. *Nature Biotechnology*, *35*(4), 316–319. https://doi.org/10.1038/nbt.3820

Docker, Inc. (2025). *Docker*. https://github.com/docker

Eizenga, J. M., Novak, A. M., Sibbesen, J. A., Heumos, S., Garrison, E., Sirén, J., & Paten, B. (2020). Pangenome graphs. *Annual Review of Genomics and Human Genetics*, *21*, 139–162.

Garrison, E., Sirén, J., Novak, A. M., Hickey, G., Eizenga, J. M., Dawson, E. T., Jones, W., Garg, S., Markello, C., Lin, M. F., Paten, B., & Durbin, R. (2018). Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature Biotechnology*, *36*(9), 875–879.

GENCODE. (2024). *Comprehensive gene annotation (ALL) GTF file, release 47 (GRCh38.p14)*. https://www.gencodegenes.org/human/

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. Ch., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, *101*(23), e215–e220. https://doi.org/10.1161/01.CIR.101.23.e215

Hasin, Y., Seldin, M., & Lusis, A. (2017). Multi-omics approaches to disease. *Genome Biology*, *18*(1), 83. https://doi.org/10.1186/s13059-017-1215-1

Heath, A. P., Ferretti, V., Agrawal, S., An, M., Angelakos, J. C., Arya, R., Bajari, R., Baqar, B., Barnowski, J. H., Burt, J.others. (2021). The NCI genomic data commons. *Nature Genetics*, *53*(3), 257–262.

Hu, J. K., Wang, X., & Wang, P. (2014). Testing gene-gene interactions in genome wide association studies. *Genetic Epidemiology*, *38*(2), 123–134. https://doi.org/10.1002/gepi.21786

Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3*, 160035. https://doi.org/10.1038/sdata.2016.35

Johnson, Alistair, Pollard, T., & Mark, R. (2016). *MIMIC-III clinical database (version 1.4)*. PhysioNet. https://doi.org/10.13026/C2XW26

Johnson, A., Pollard, T., & Mark, R. (2022). *MIMIC-III clinical database CareVue subset (version 1.4)*. PhysioNet. https://doi.org/10.13026/8a4q-w170

Koenig, Z.others. (2023). *A harmonized public resource of deeply sequenced diverse human genomes*. https://doi.org/10.1101/2023.01.23.525248

LeMieux, J. (2025). Protein-protein interactions get a new groove on. *Genetic Engineering & Biotechnology News (GEN)*. https://www.genengnews.com/insights/protein-protein-interactions-get-a-new-groove-on/

Lenski, R. E. (2001). Twice as natural. *Nature*, *414*(6861), 255. https://doi.org/10.1038/35104715

Lv, T., Zhang, Y., Liu, J., Kang, Q., & Liu, L. (2024). Multi-omics integration for both single-cell and spatially resolved data based on dual-path graph attention auto-encoder. *Briefings in Bioinformatics*, *25*(5), bbae450. https://doi.org/10.1093/bib/bbae450

Merkel, D. (2014). Docker: Lightweight linux containers for consistent development and deployment. *Linux Journal*, *2014*(239), Article 2.

Misra, B. B., Langefeld, C., Olivier, M., & Cox, L. A. (2019). Integrated omics: Tools, advances and future approaches. *Journal of Molecular Endocrinology*, *62*(1), R21–R45. https://doi.org/10.1530/JME-18-0055

Moore, R., Casale, F. P., Bonder, M. J., Horta, D., Franke, L., Barroso, I., & Stegle, O. (2018). A linear mixed-model approach to study multivariate gene–environment interactions. *Nature Genetics*, *50*(7), 1167–1174.

National Institutes of Health. (2023, May). *Scientists release a new human "pangenome" reference*. National Institutes of Health. https://www.nih.gov/news-events/news-releases/scientists-release-new-human-pangenome-reference

Olivier Delaneau. (n.d.). *GLIMPSE2 tutorial*. https://odelaneau.github.io/GLIMPSE/docs/tutorials/getting_started/.

Piñero, J., Ramírez-Anguita, J. M., Saüch-Pitarch, J., Ronzano, F., Centeno, E., Sanz, F., & Furlong, L. I. (2020). The DisGeNET knowledge platform for disease genomics: 2019 update. *Nucleic Acids Research*, *48*(D1), D845–D855.

Pontén, F., Jirström, K., & Uhlen, M. (2008). The human protein atlas—a tool for pathology. *The Journal of Pathology*, *216*(4), 387–393. https://doi.org/https://doi.org/10.1002/path.2440

Purcell, S., & Chang, C. (n.d.). *PLINK v2.0*. www.cog-genomics.org/plink/2.0/.

PyTorch contributors. (2016--2025). *PyTorch*. https://pytorch.org.

Ramsey, J. D., Zhang, K., Glymour, M., Romero, R. S., Huang, B., Ebert-Uphoff, I., & Glymour, C. (2018). TETRAD—a toolbox for causal discovery. *8th International Workshop on Climate Informatics*.

Rubinacci, S.others. (2021). Efficient phasing and imputation of low-coverage sequencing data using large reference panels. *Nature Genetics*, *53*(1), 120–126. https://doi.org/10.1038/s41588-020-00756-0

Rubinacci, S.others. (2023). Imputation of low-coverage sequencing data from 150,119 UK biobank genomes. *Nature Genetics*, *55*(7), 1088–1090. https://doi.org/10.1038/s41588-023-01438-3

Samtools. (2025). *BCFtools manual*. https://samtools.github.io/bcftools/

Suhre, K., Shin, S.-Y., Petersen, A.-K., Mohney, R. P., Meredith, D., Wägele, B., Altmaier, E., CARDIoGRAM, Deloukas, P., Erdmann, J., Grundberg, E., Hammond, C. J., Angelis, M. Hr. de, Kastenmüller, G., Köttgen, A., Kronenberg, F., Mangino, M., Meisinger, C., Meitinger, T., … Gieger, C. (2011). Human metabolic individuality in biomedical and pharmaceutical research. *Nature*, *477*(7362), 54–60.

Sun, B. B., Chiou, J., Traylor, M., Benner, C., Hsu, Y., Richardson, T., Surendran, P., Mahajan, A., Robins, C., Vasquez-Grinnell, S., Hou, L., Kvikstad, E., Burren, O., Davitte, J., Ferber, K., Gillies, C., Hedman, Å., Hu, S., Lin, T., … Whelan, C. (2023). Plasma proteomic associations with genetics and health in the UK biobank. *Nature*. https://doi.org/10.1038/s41586-023-06592-6

Sun, Benjamin B., Maranville, J. C., Peters, J. E., Stacey, D., Staley, J. R., Blackshaw, J., Burgess, S., Jiang, T., Paige, E., Surendran, P., Oliver-Williams, C., Kamat, M. A., Prins, B. P., Wilcox, S. K., Zimmerman, E. S., Chi, A., Bansal, N., Spain, S. L., Wood, A. M., … Butterworth, A. S. (2018). Genomic atlas of the human plasma proteome. *Nature*, *558*(7708), 73–79.

Szklarczyk, D., Gable, A. L., Nastou, K. C., Lyon, D., Kirsch, R., Pyysalo, S., Doncheva, N. T., Legeay, M., Fang, T., Bork, P.others. (2021). The STRING database in 2021: Customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Research*, *49*(D1), D605–D612.

Tommaso, P. D., Chatzou, M., Floden, E., Barja, P. P., Palumbo, E., & Notredame, C. (2017). *Nextflow enables reproducible computational workflows*. https://github.com/nextflow-io/nextflow

Wang, Q. S., Hasegawa, T., Namkoong, H.others. (2024). Statistically and functionally fine-mapped blood eQTLs and pQTLs from 1,405 humans reveal distinct regulation patterns and disease relevance. *Nature Genetics*, *56*, 2054–2067. https://doi.org/10.1038/s41588-024-01896-3

Wharrie, S., Yang, Z., Raj, V., Monti, R., Gupta, R., Wang, Y., Martin, A., O'Connor, L. J., Kaski, S., Marttinen, P., Palamara, P. F., Lippert, C., & Ganna, A. (2022). *HAPNEST synthetic dataset*. BioStudies, S-BSST936. https://www.ebi.ac.uk/biostudies/studies/S-BSST936

Wojcik, G. L., Graff, M., Nishimura, K. K.others. (2019). Genetic analyses of diverse populations improves discovery for complex traits. *Nature*, *570*, 514–518. https://doi.org/10.1038/s41586-019-1310-4

Zhang, B., Biddanda, A., Gunnarsson, Á. F., Cooper, F., & Palamara, P. F. (2023). Biobank-scale inference of ancestral recombination enables genealogical analysis of complex traits. *Nature Genetics*, *55*, 768–776. https://doi.org/10.1038/s41588-023-01379-x

Zimmer, C. (2016). *As DNA reveals its secrets, scientists are assembling a new picture of humanity*. https://www.statnews.com/2016/10/07/dna-genome-sequencing-new-maps/