Probabilistic Ancestral Inference from Incomplete Genetic Data

Makenna Worley

October 2025

Abstract

This project explores whether probabilistic machine-learning models can reconstruct missing ancestral genotypes and predict hereditary trait risks from incomplete genetic data. Using *Drosophila melanogaster* (fruit fly) as a model organism [1, 2], this project will simulate Mendelian inheritance patterns across generations using population-genetic frameworks such as `simuPOP` and `msprime` [3, 4]. It will apply probabilistic and machine-learning methods, including Bayesian inference and neural-network architectures, to infer missing ancestral genotypes [5, 6].

The system will be tested for accuracy across independent simulation datasets, validated using statistical metrics, and visualized through an interactive dashboard. The end goal is to develop a proof-of-concept that demonstrates whether probabilistic inference and machine learning can recover genetic information in both research and applied contexts such as conservation biology and medicine.

1 Introduction

Incomplete ancestry data remains one of the most persistent barriers in modern genetics, constraining our ability to reconstruct inheritance patterns, predict hereditary traits, and model population history with confidence. Sequencing every ancestor is rarely feasible due to cost, data degradation, or ethical limitations [7, 8, 9].

This project aims to develop a computational method capable of reconstructing ancestral genotypes and predicting traits based on partial data. Using simulated Drosophila melanogaster (fruit fly) populations—whose inheritance patterns follow well-established Mendelian rules—this project will investigate whether probabilistic models can reliably infer missing genetic data [1, 2]. Drosophila provides an especially strong foundation for simulation because its genome has been fully characterized and extensively quantified through resources such as the Drosophila Genetic Reference Panel (DGRP), which offers well-validated allele frequencies and population-level variation critical for controlled modeling [1]. The simulation framework will employ population-genetic tools such as `simuPOP` and `msprime`, which enable controlled forward-time and coalescent-based simulations [3, 4].

The core objectives are:

- To design a simulation and inference system that models genetic inheritance under uncertainty.
- To implement, train, and test probabilistic and machine-learning models for genotype reconstruction [5, 6].
- To develop an interactive web dashboard for visualization, evaluation, and comparison of model results.

If successful, this project would establish a reproducible methodology for genotype reconstruction under uncertainty, offering a new computational tool for studies that rely on incomplete or partially observed genetic data.

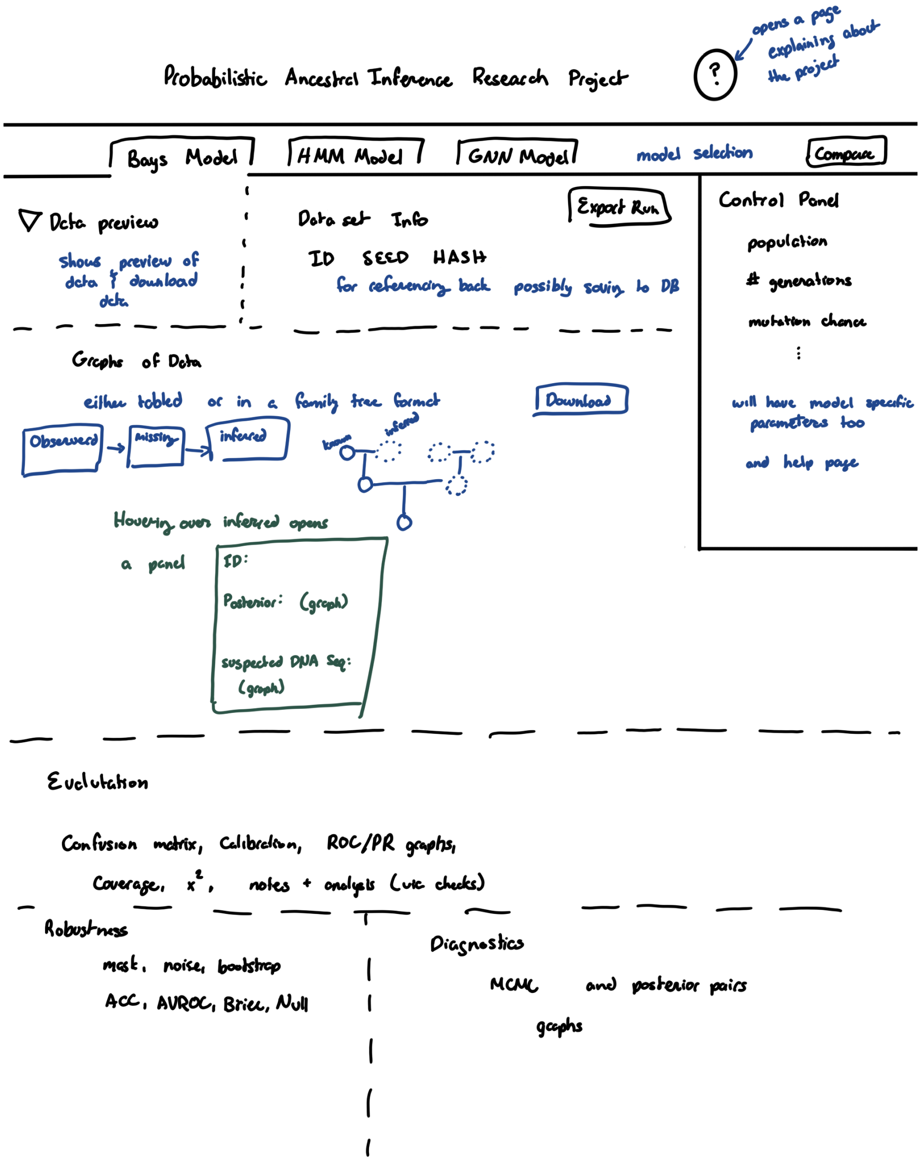

Figure 1: I plan on using a similar design language to my 3D Visualizer.

2 Background

2.1 Scientific Context

Reconstructing ancestral genotypes from limited data is an ongoing challenge in computational genetics [7, 8, 9]. Methods such as Bayesian inference, Hidden Markov Models (HMMs), and graph-based models offer complementary strengths for modeling uncertainty and inheritance linkage across generations [5, 10, 6].

Table 1: Comparison of Probabilistic Inference Methods

Method	**Strengths**	**Limitations**	**Best Used When**
Bayesian Inference	Interpretable; incorporates biological priors; models uncertainty explicitly.	High computational cost; slower convergence for large datasets.	Prior biological knowledge is available and interpretability is important.
Hidden Markov Model (HMM)	Efficient for sequential linkage modeling; strong theoretical foundation in genetics.	Simplifies non-linear inheritance; assumes independence between hidden states.	Modeling recombination or chromosomal linkage across loci.
Graph-Based Model	Captures multi-way relationships and complex inheritance structures; scalable to large populations.	Requires large datasets and careful tuning; less interpretable than Bayesian models.	Modeling population-level inheritance or highly connected genetic networks.

However, many existing inference approaches rely on dense or fully observed datasets, which limits their applicability in real-world genetic studies. Traditional maximum-likelihood and coalescent-based methods (e.g., STRUCTURE, IMa2, BEAST) assume complete genotype data or well-defined population priors, making them sensitive to missing information [7, 8].

Similarly, HMM-based linkage and recombination tools such as *fastPHASE* and *Beagle* perform well for contiguous genotype data but degrade when ancestral nodes are unobserved or sparsely sampled [10]. Pedigree reconstruction algorithms like *COLONY* and *FRANz* can infer parentage relationships, yet they depend on extensive genotyping coverage and are computationally expensive at scale [9].

These limitations motivate the development of models that remain robust under uncertainty and partial observability—particularly in contexts where sequencing every ancestor is infeasible due to cost, degradation, or ethical constraints.

Unlike existing approaches that depend on dense or real-world datasets, this project leverages *simulated inheritance data* from *Drosophila melanogaster* to create a fully controlled testing environment. Fruit flies provide an ideal foundation for simulation—their genomes are fully mapped, inheritance patterns are predictable, and generations can be modeled computationally at scale [2, 1, 4]. By masking genotypes at varying rates within these simulated populations, the framework systematically evaluates how different probabilistic models (Bayesian, HMM-based, and graph-based) perform under increasing levels of missing information. This design enables direct comparison of model robustness and scalability while avoiding the confounding biological noise and sampling bias that affect empirical datasets.

In this project, simulations will focus on a subset of key loci rather than the entire *Drosophila melanogaster* genome. Each simulated individual will carry diploid genotypes at multiple independent loci representing autosomal regions with well-characterized Mendelian inheritance. These loci will be parameterized using empirically derived allele frequencies and recombination rates drawn from the Drosophila Genetic Reference Panel (DGRP) [1].

Whole-genome simulation is computationally possible but unnecessary for evaluating probabilistic inference performance; instead, the selected loci will be designed to capture essential genetic processes—such as segregation, recombination, and mutation—without the overhead of full-scale genomic complexity.

Simulation frameworks such as `simuPOP` and `msprime` will be used to generate multi-generational populations following standard Mendelian rules [3, 4]. Masking of genotypes will then be applied at controlled rates to emulate missing ancestral data. This structure allows the models to focus on inferring inheritance patterns and reconstructing ancestral genotypes rather than fitting species-specific genome architecture.

2.2 Technological Context

Recent advances in AI and simulation frameworks make this project technically feasible and computationally efficient. Libraries such as `simuPOP` and `msprime` support large-scale, biologically grounded genetic simulations that have been rigorously validated by biologists [3, 4]. Together, they enable fine-grained control over inheritance, recombination, and mutation parameters while maintaining scalability across generations.

The simulation pipeline will integrate both tools: `simuPOP` will model forward-time population dynamics, while `msprime` will generate coalescent-based ancestry data to ensure statistical realism. The resulting datasets will provide a reproducible foundation for probabilistic inference and model validation.

For statistical modeling, the project will employ PyMC (built on modern PyTensor) to implement the Bayesian framework, using biologically informed priors for allele frequencies and inheritance probabilities [5]. The Hidden Markov Model (HMM) will be developed using either hmmlearn or pomegranate, both of which offer flexible APIs for representing sequential genotype transitions and recombination events [10].

The optional graph-based approach will utilize PyTorch Geometric to represent individuals and genetic relationships as nodes and edges within a dynamic inheritance graph [6]. Model training and evaluation outputs will be served through a FastAPI backend and visualized using a custom React dashboard, allowing for interactive exploration of genotype predictions, uncertainty, and accuracy metrics. The complete system will be containerized with Docker to ensure reproducibility, portability, and modular deployment [11, 12].

These tool choices were made for their open-source accessibility, active research communities, and demonstrated performance in genetics, simulation, and AI applications.

I also bring prior experience with several components of this stack: I have prior experience with React from the CircuitCraft summer research program, where I helped build an interactive circuit-drawing application using ReactFlow. I also used FastAPI in the MVP of my differential equations project, which integrated mathematical modeling with a web-based backend. In addition, I am currently validating my RSQL database for integration into this system, ensuring that data handling and deployment components are fully prepared for the development phases of this project.

Through this project, I aim to extend my skills from general web and data engineering into advanced computational genetics—specifically, probabilistic modeling, simulation design, and model interpretability. I will need to learn how to implement Bayesian and Hidden Markov Models in practice, interpret probabilistic outputs within a biological context, and calibrate genetic simulations using tools such as simuPOP and msprime. Developing these skills will challenge me to connect theoretical machine learning with applied biological data, bridging the gap between data science and computational biology.

3 Proposed Work

3.1 Proof of Concept

3.1.1 Overview

The core proof of concept (PoC) for this project will be a working system capable of reconstructing missing ancestral genotypes from simulated *Drosophila melanogaster* populations using probabilistic inference [1, 2]. At minimum, the PoC will demonstrate that a

Bayesian model trained on partially masked simulation data can accurately recover missing genotypes and visualize results through an interactive dashboard. This deliverable will establish both the technical and scientific feasibility of the broader goal: probabilistic ancestral inference from incomplete genetic datasets.

3.1.2 Specific Tasks

1. Simulate multi-generational fruit-fly populations with known inheritance rules using `simuPOP` and `msprime` [3, 4].
2. Develop and train a baseline Bayesian model to infer missing ancestral genotypes [5].
3. Generate three independent datasets (training, validation, and testing) to ensure statistical robustness.
4. Implement a lightweight `Streamlit` dashboard for visualization and performance monitoring.
5. Evaluate inference accuracy using chi-square and likelihood-based metrics.

3.1.3 Rationale

The proof of concept focuses on reconstructing masked genotypes in simulated ancestry data rather than predicting real-world phenotypes. This controlled environment allows for quantitative validation against known ground-truth data and clear visualization of uncertainty. Demonstrating that a Bayesian model can successfully recover missing genetic information will validate the feasibility of the inference framework before expanding to additional model types and real-world data.

3.1.4 Deployment

The proof of concept will be deployed through a lightweight `Streamlit` dashboard to ensure ease of demonstration and clear visualization of results. All simulation data will be stored as `.csv` files and loaded directly into the dashboard alongside the Bayesian model implemented in Python. The deployment will provide a simple local interface for loading datasets, running inference, and viewing performance metrics, requiring no external infrastructure. This streamlined approach ensures that the PoC can be demonstrated reliably, reused in later development phases, and extended to incorporate additional models as the project expands.

3.2 Full Project Development

3.2.1 Specific Tasks

1. Extend the simulation pipeline to support larger populations and additional inheritance parameters using `simuPOP` and `msprime` [3, 4].

2. Implement multiple probabilistic models (Bayesian, HMM-based, and optionally graph neural networks) to predict missing ancestor genotypes [5, 10, 6].

3. Regenerate training, validation, and testing datasets under varying missingness conditions for comparative analysis.

4. Integrate a `FastAPI` backend for model serving and data management [11].

5. Build a full `React` dashboard for interactive visualization and performance comparison [12].

6. Validate models using chi-square, likelihood-ratio, and additional evaluation metrics.

7. Package and containerize the complete system using `Docker` for reproducibility and portability.

3.2.2 Rationale

Genetic datasets are frequently incomplete, yet probabilistic models excel at reasoning under uncertainty [5, 10]. Using simulated data provides complete control over inheritance, mutation, and masking parameters, enabling reproducible experiments and rigorous validation [3, 4]. The choice of *Drosophila melanogaster* minimizes biological ambiguity while providing a scalable, well-characterized system for testing computational approaches [1, 2]. Employing three independent datasets reduces overfitting risk, and the `FastAPI--React` architecture ensures long-term scalability and maintainability [11, 12]. Together, these design decisions align with best practices in research software development, offering a clear path from prototype to deployable system.

3.3 Plan of Work

1. Simulation and Data Preparation: Establish a pipeline for simulating population genetics data under Mendelian inheritance using `simuPOP` and `msprime` [3, 4]. Define masking rates and mutation probabilities to represent incomplete ancestry. Verify reproducibility through seed-based runs and export datasets in standardized formats.

2. Model Design and Training: Implement and compare model types — Bayesian, HMM-based probabilistic inference, and neural-network–based graph learning [5, 10, 6]. Optimize runtime for larger simulations, improve the visualization layout, and prepare full documentation for reproducibility.

3. Validation and Statistical Analysis: Introduce controlled noise and missingness in datasets to evaluate model robustness. Perform chi-square and likelihood-ratio tests across multiple conditions, recording performance metrics and error bounds.

4. Dashboard and Visualization: Develop an interactive web interface using `React` for visualizing genotype predictions, accuracy metrics, and confidence intervals [12]. Integrate `FastAPI` endpoints to dynamically query data [11].

5. Refinement and Documentation: Evaluate usability and performance bottlenecks. Optimize runtime for larger simulations, improve the visualization layout, and prepare full documentation for reproducibility.

Absolute Minimum: a working simulation and baseline Bayesian model with a `Streamlit` dashboard for reporting out statistics.

Expected: all the features of Absolute Minimum and the addition of a full validation pipeline with custom `React` dashboard integration, making use of two models—Bayesian and HMM-based.

Aspirational: all the features of Expected plus the addition of a graph-based model with comparison statistics showing the effectiveness of all models.

3.4 Preliminary Work

Initial research and AI-supported brainstorming identified effective toolchains (`FastAPI`, `React`, `simuPOP`, `msprime`) and validation strategies using independent datasets [3, 4, 11, 12].

4 Timeline

Table 2: Project Timeline

Phase	Dates	Time est.	Focus	Deliverables
Phase 0: Preparation	Oct 2025	20 hrs	Validate `RSQL` database and build mini-projects using `simuPOP` and `msprime` to establish familiarity with both libraries [3, 4].	Verified `RSQL` functionality and working demo scripts for both simulators.
Phase 1: Proof of Concept	Nov 2025	70 hrs	Develop genetic simulation pipeline and baseline Bayesian probabilistic model; generate and handle simple datasets [5].	Working simulation, MVP model achieving $\geq$70% inference accuracy, and `Streamlit` dashboard.
Phase 2: Expansion	Dec–Jan 2026	90 hrs	Add HMM-based statistical model; handle larger, more complex datasets; perform validation via chi-square, likelihood-ratio, and evaluation metrics [10].	Python-based system with validation metrics and visualizations in `Streamlit`.
Phase 3: Web Application	Feb–Mar 2026	70 hrs	Develop `FastAPI` backend and connect with `React` dashboard; integrate and secure `RSQL` database; refine visualizations. Possibly implement aspirational features [11, 12].	Functional web dashboard with validation metrics and visualizations (migrated from `Streamlit` to `React`).
Phase 4: Code Freeze	Apr 2026	40 hrs	Finalize dashboard design language, optimize performance, and prepare documentation.	Containerized final build and comprehensive report for presentation.

Total estimated effort: approximately 290 hours.

5 Evaluation and Conclusion

Evaluation will focus on:

- Accuracy: Comparing predicted vs. known genotypes using statistical metrics (precision, recall, chi-square) [10, 5].

- Performance: Measuring runtime efficiency and resource usage.
- Usability: Ensuring the dashboard clearly communicates results and model uncertainty [11, 12].
- Impact: Considering ethical implications (interpretability) and relevance for research and conservation [1, 2].
  - Use of simulated data avoids privacy risks but informs future work on real genomic datasets.
  - Emphasis on interpretability to prevent misuse or overconfidence in probabilistic predictions [5].
  - Applications in conservation must avoid deterministic or biased interpretations of genetics [1].

The expected outcome is a validated prototype demonstrating that probabilistic models can effectively recover missing genetic information. This will establish groundwork for future research in population genetics and applied inference systems [3, 4].

Conclusion

Incomplete or sparsely sampled ancestry data limits the ability of geneticists to reconstruct inheritance patterns, predict hereditary traits, and model population history with confidence. This project addresses that challenge by developing a simulation-driven framework that evaluates whether probabilistic inference—specifically Bayesian models, Hidden Markov Models, and graph-based methods—can recover missing ancestral genotypes using controlled *Drosophila melanogaster* data. By leveraging `simuPOP` and `msprime` to generate reproducible, ground-truth datasets, the system will enable rigorous testing of how well different probabilistic models perform under uncertainty.

The proposed solution integrates genetic simulation, statistical modeling, and interactive visualization to create a reproducible toolset for understanding how incomplete data affects genotype inference. If successful, this project will deliver a validated prototype demonstrating that probabilistic models can accurately reconstruct missing genetic information, even when ancestral nodes are unobserved or sparsely sampled.

Such a result would provide a methodological foundation for future work in computational genetics, supporting applications in population modeling, conservation biology, and biomedical trait prediction. More broadly, this project will contribute a scalable, interpretable, and ethically grounded approach to reasoning under genetic uncertainty—offering new possibilities for research settings where complete genomic data is unattainable.

6 References

References

[1] Mackay, T. F. C., Richards, S., Stone, E. A., et al. (2012). *The Drosophila melanogaster Genetic Reference Panel. Nature, 482*(7384), 173–178.

[2] Adams, M. D., Celniker, S. E., Holt, R. A., et al. (2000). *The genome sequence of Drosophila melanogaster. Science, 287*(5461), 2185–2195.

[3] Peng, B., & Kimmel, M. (2005). *simuPOP: A forward-time population genetics simulation environment. Bioinformatics.*

[4] Kelleher, J., Etheridge, A. M., & McVean, G. (2018). *Efficient coalescent simulation and genealogical analysis for large sample sizes. PLOS Computational Biology, 14*(5), e1006581.

[5] Salvatier, J., Wiecki, T., & Fonnesbeck, C. (2016). *Probabilistic programming in Python using PyMC3. PeerJ Computer Science.*

[6] Paszke, A., et al. (2019). *PyTorch: An imperative style, high-performance deep learning library. NeurIPS.*

[7] Pritchard, J. K., Stephens, M., & Donnelly, P. (2000). *Inference of population structure using multilocus genotype data. Genetics, 155*(2), 945–959.

[8] Hey, J., & Nielsen, R. (2004). *Multilocus methods for estimating population sizes, migration rates and divergence time, with applications to the divergence of Drosophila pseudoobscura and D. persimilis. Genetics, 167*(2), 747–760.

[9] Wang, J. (2012). *Computationally efficient sibship and parentage assignment from multilocus marker data. Genetics, 191*(1), 183–194.

[10] Browning, S. R., & Browning, B. L. (2007). *Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. The American Journal of Human Genetics, 81*(5), 1084–1097.

[11] FastAPI Documentation (2024). *Modern web framework for building APIs with Python 3.7+.*

[12] React Documentation (2024). *React: A JavaScript library for building user interfaces.*