Molecular Biology and Evolution, 2026, pp. 1–12
DOI added during production
Published: Date added during production
Paper

PopGenAgent: Tool-Aware, Reproducible, Report-Oriented Workflows for Population Genomics

Houcheng Su,[1] Weicai Long,[1] Juning Feng,[1] Yusen Hou[1] and Yanlin Zhang[1,*]

[1]Information Hub, Hong Kong University of Science and Technology (Guangzhou), Guangzhou, 511400, Guangdong, China

[*]Corresponding author. yanlinzhang@hkust-gz.edu.cn

Abstract

Population-genetic inference routinely requires coordinating many specialized tools, managing brittle file formats, iterating through diagnostics, and converting intermediate results into interpretable figures and written summaries. Although workflow frameworks improve reproducibility, substantial last-mile effort remains for parameterization, troubleshooting, and report preparation. Here we present PopGenAgent, a turnkey, report-oriented delivery system that packages a curated library of population-genetics toolchains into validated execution and visualization templates with standardized I/O contracts and full provenance capture. PopGenAgent separates retrieval-grounded user assistance for interpretation and write-up from conservative, template-driven execution that emphasizes auditable commands, artefact integrity checks, and report-ready figure generation. To control operating cost, an economical language model is used for template selection, parameter instantiation, and minor repairs, while higher-capacity models can be invoked selectively for narrative report generation grounded in recorded artefacts. We evaluate PopGenAgent on a broad panel of routine and advanced tasks spanning preprocessing, population structure analysis, and allele-sharing statistics, and we further demonstrate end-to-end replication of standard analyses on 26 populations from the 1000 Genomes Project, reproducing canonical summaries including ROH/heterozygosity profiles, LD decay, PCA, ADMIXTURE structure, TreeMix diagnostics, and f-statistics. Together, these results indicate that a validated template library coupled with provenance-aware reporting can substantially reduce manual scripting and coordination overhead while preserving reproducibility and step-level inspectability for population-genomic studies.

Keywords Population Genetics, Large Language Models, Computational Genomics, Multi-agent Agents, 1000 Genomes Project

Abbreviations PopGenAgent, Population Genomics Workflows, multi-agent system

Introduction

Population genetics provides a foundational set of quantitative tools for studying genetic diversity, population structure, and evolutionary history Zhou et al. (2023). Despite mature theory and widely adopted software, practical population-genomic analysis remains labor-intensive: researchers must coordinate many command-line tools with heterogeneous file formats, tune parameters across dependent steps, diagnose failures or artefact inconsistencies, and ultimately convert raw outputs into interpretable figures and written summaries suitable for downstream modeling and reporting Kamvar et al. (2015).

To reduce technical barriers, the community has developed reproducible workflow frameworks and curated pipelines Wratten et al. (2021). Resources such as nf-core Ewels et al. (2020) and GenPipes Bourgey et al. (2019) standardize execution environments and improve reproducibility. However, most pipeline solutions remain intrinsically *workflow-centric* rather than *analysis-centric*: users still perform substantial manual work when iterating across alternative preprocessing decisions, exploring complementary structure/statistics analyses, or troubleshooting format- and assumption-sensitive steps. In addition, report construction typically remains external to the pipeline, requiring manual figure generation, provenance bookkeeping, and narrative write-up—a major source of time cost and irreproducibility.

Large language models (LLMs) offer a complementary interface for reducing scripting overhead and supporting interactive analysis Zhao et al. (2023); Chang et al. (2024); Kasneci et al. (2023). In computational biology, early LLM-enabled systems have demonstrated tool calling and workflow assistance in domains such as genome analysis and transcriptomics Huang et al. (2025); Su et al. (2025); Zhou et al. (2024). For population genetics, however, the key practical requirement is not "agentic novelty" but *turnkey delivery*: reliable execution of widely used toolchains under real-world failure modes, production of publication-oriented figures, and generation of concise, inspectable reports grounded in the actual outputs and full provenance. A one-shot script generator is insufficient when conclusions depend on stepwise inspection (e.g., structure inference, summary statistics, diagnostic checks) and when upstream artefact issues must be detected before propagating downstream Kopelman et al. (2015); Li and Liu (2018); Leigh et al. (2015).

Here we present **PopGenAgent**, a practical delivery system for population-genetic analyses that couples a curated library of validated execution-and-visualization templates for common

Received: Date Month Year. **Revised received:** Date Month Year. **Accepted:** Date Month Year
© The Author(s) 2026. Copyright and licence statements to be updated by the publisher during production.

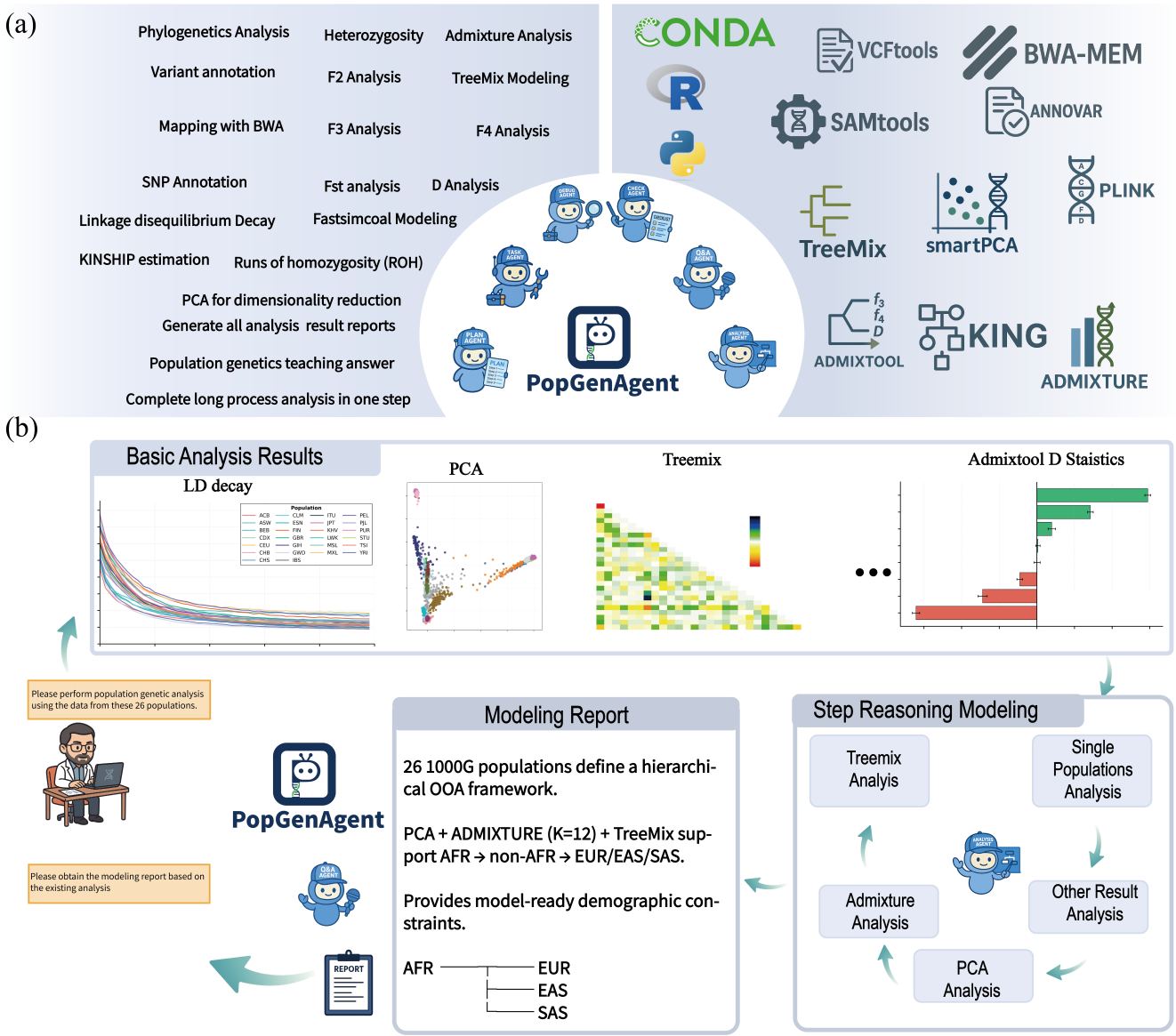


Figure 1 Overview of PopGenAgent functionalities and report-oriented interactive analysis workflow. **(a)** Summary of population-genetic tasks delivered by PopGenAgent and the external software it orchestrates, spanning read alignment and variant processing (e.g., BWA-MEM, SAMtools, VCFtools, ANNOVAR), core population-genetic analyses (e.g., PLINK, smartPCA, ADMIXTURE, TreeMix, ADMIXTOOLS), and optional demographic-modeling interfaces (e.g., `fastsimcoal2`, `demes`). **(b)** Example end-to-end workflow driven by natural-language requests. PopGenAgent executes a sequence of standard analyses (e.g., LD decay, F_{ST}, PCA, ADMIXTURE, TreeMix, and allele-sharing statistics) and produces report-ready figures and intermediate artefacts. These outputs are then consolidated into a modeling-oriented report that summarizes population structure and highlights model-relevant constraints (e.g., candidate groupings and admixture hypotheses), supporting iterative, user-steered follow-up analyses with full provenance.

population-genetics tasks with a report-oriented interactive interface. PopGenAgent executes standard toolchains (e.g., PLINK, ADMIXTURE, smartPCA, TreeMix, ADMIXTOOLS) with explicit I/O contracts, conservative defaults, and plotting scripts, while continuously recording provenance (commands, parameters, software versions, logs, and artefacts) to support auditing and reproducibility. To reduce manual effort in interpretation and write-up, PopGenAgent further provides session-aware Q&A backed by a retrieval-augmented knowledge base of population-genetics literature and methodological resources, enabling users to query tool assumptions and explain outputs in context, and to draft report text aligned with the produced figures. Operationally, PopGenAgent is deployed as an interactive web system that manages multiple analysis sessions, supports concurrent execution, and provides controls for pausing, resuming, and interrupting runs, thereby enabling iterative refinement without rebuilding pipelines from scratch.

We evaluate PopGenAgent from two complementary perspectives that reflect routine population-genetics practice. First, we quantify end-to-end executability across a panel of preprocessing and analysis tasks by repeating each task five times and recording whether the expected outputs and figures are produced without manual rewriting of generated commands. Second, we assess robustness and

practica l coverage by replicating a standard suite of analyses on the 1000 Genomes Project dataset, including ROH and heterozygosity summaries, LD decay, PCA, ADMIXTURE, TreeMix diagnostics, and allele-sharing statistics, and by assembling these outputs into modeling-oriented reports grounded in full provenance. Together, these experiments test whether PopGenAgent can reduce manual scripting and reporting overhead while preserving inspectability and reproducibility for population-genomic research.

Results

Execution of Population-Genetics Toolchains

PopGenAgent was evaluated as a practical delivery system for population-genetic analyses by testing its ability to execute a broad panel of routine and advanced tasks end-to-end, including read mapping, variant processing, quality control, and downstream population-structure and statistics workflows (Table 1). In addition to the benchmarked tasks summarized in the main text, PopGenAgent maintains a broader template-packaged tool stack spanning core processing/QC, population structure, diversity/LD summaries, admixture tests, and graph-based modeling; the complete validated inventory is provided in the Appendix. Across five independent runs per task, all benchmarked workflows completed successfully in all five runs without manual rewriting of the generated command lines. This repeated-run benchmark covered widely used software packages, including BWA Li (2013); Li and Durbin (2010, 2009), SAMtools Danecek et al. (2021), ANNOVAR Wang et al. (2010), VCFtools Danecek et al. (2011), PLINK Chang et al. (2015), KING Manichaikul et al. (2010), smartPCA Galinsky et al. (2016a,b), ADMIXTURE Alexander et al. (2009), TreeMix Pickrell and Pritchard (2012), and ADMIXTOOLS Maier et al. (2023).

This complete repeated-run success indicates stable executability not only for core preprocessing stages, such as read mapping, variant filtering, SNP annotation, and standard genotype/VCF filtering, but also for more configuration-sensitive downstream analyses, including LD pruning, PCA, ADMIXTURE, TreeMix, and f-statistics. The results therefore support the conclusion that, after refinement of the task-specific knowledge base and execution templates, PopGenAgent can reliably deliver the expected output artefacts and figures across the benchmarked population-genetics workflows.

PopGenAgent is designed for usability under real-world failure modes rather than as a one-shot script generator. During execution, the system produces an explicit, editable plan and continuously records provenance (commands, parameters, software versions, logs, and artefacts) to support step-level auditing and reproducibility. When errors occur, PopGenAgent surfaces concise diagnostics and performs bounded, template-level recovery (planning/execution/debugging/validation) following the BioMaster execution kernel Su et al. (2025); when automatic recovery is insufficient, users can pause execution, revise step-level scripts through the web interface (Figure 4), and resume from the point of interruption. A lightweight validation stage checks intermediate artefacts (e.g., existence and basic integrity) to reduce the risk of silent failures propagating into downstream inference.

A key deliverable is report-ready output. Upon completion, PopGenAgent generates publication-oriented figures and a one-click report that aggregates key results together with full provenance; session-aware Q&A further supports in-context interpretation of standard outputs (e.g., PCA projections, ADMIXTURE cross-validation curves, TreeMix residuals, and ROH/heterozygosity summaries). To reduce operating cost while preserving readability, an economical LLM is used for template selection, parameter instantiation, and minor repairs, whereas a higher-capacity model can be invoked selectively for report narration and figure-level explanations grounded in the recorded artefacts.

Finally, PopGenAgent is extensible in a way that prioritizes user effort reduction. New tools or customized workflows can be incorporated by adding minimal tool descriptors and validated execution/visualization templates (expected inputs/outputs, invocation patterns, and plotting/report recipes), enabling the system to deliver additional analyses as reproducible, report-ready modules. As an integration validation case, we additionally incorporated `gpmap-tools` Martí-Gómez et al. (2026) to test that newly published software can be packaged under the same template and reporting framework. The main text lists representative benchmarked tools, whereas the complete validated tool inventory is provided in the Appendix, and per-tool configuration knowledge is available in the public code repository.

Interactive Analysis with Retrieval-Grounded Assistance and One-Click Modeling Reports

PopGenAgent is designed as a practical, end-to-end delivery system for population-genetics analyses, emphasizing usability and report-ready outputs rather than agentic novelty. In routine use, users provide an analysis objective and data, and PopGenAgent executes a validated sequence of population-genetics tool steps, continuously collecting intermediate artefacts and synthesizing them into a concise modeling-oriented report (Figure 1b). This stepwise workflow integrates common population-structure and summary-statistic analyses—including PCA, ADMIXTURE, TreeMix, LD-decay, ROH/heterozygosity summaries, and allele-sharing tests—so that intermediate results can directly inform subsequent modeling choices (e.g., candidate groupings, plausible admixture hypotheses, and diagnostic checks), while remaining fully traceable to the generated outputs.

To reduce manual effort in interpretation and write-up, PopGenAgent provides a retrieval-grounded Chat interface that links user queries to an up-to-date knowledge base of population-genetics literature and methodological resources. During and after execution, the Chat interface explains tool assumptions and typical output patterns (e.g., PCA projections, ADMIXTURE cross-validation behavior and ancestry profiles, TreeMix residual diagnostics, and ROH/heterozygosity summaries) and can draft report text aligned with the produced figures and tables. This design supports rapid understanding of results without requiring users to manually navigate documentation and logs.

PopGenAgent is deployed as an interactive web system that prioritizes operational convenience for real analyses (Figure 4). The interface manages multiple analysis sessions, supports concurrent task execution, and exposes controls for starting, pausing, resuming, and interrupting runs. Users can inspect intermediate artefacts as they are produced and, when necessary, intervene by revising configurations or step-level commands before continuing. Upon completion, PopGenAgent generates a one-click report that aggregates key plots and summaries together with provenance (commands, parameters, software versions, and execution

Module	Benchmarked tasks (this study)	Tool	Additional template-packaged tools
	Read mapping	BWA	
	Variant filtering	SAMtools	
	Variant calling / SNP extraction	SAMtools	
	SNP annotation	ANNOVAR	
	Basic genotype/VCF filtering	VCFtools	
	LD pruning	PLINK	
Core data processing & QC	Relatedness inference & removal	KING	Bowtie2; BCFtools; PLINK2; BEDTools
	PCA for dimensionality reduction	smartPCA	
Population structure	Model-based ancestry inference	ADMIXTURE	STRUCTURE; fastSTRUCTURE
	LD decay estimation	PLINK	
	Runs of homozygosity (ROH)	PLINK	
	Heterozygosity & inbreeding summaries	PLINK	
Diversity, relatedness & LD	Genetic distance / phylogenetic summary	PLINK	pixy; PopLDdecay
	f_2 statistics	ADMIXTOOLS	
	f_3 statistics	ADMIXTOOLS	
	f_4 statistics	ADMIXTOOLS	
	D statistics (ABBA–BABA)	ADMIXTOOLS	
Admixture tests & graph-based modeling	TreeMix modeling	TreeMix	admixtools (R); qpAdm; qpGraph; bed2treemix
Selection & simulation	–	–	selscan; Relate; CLUES; msprime; SLiM
Extensibility validation	–	–	gpmap-tools (integration validation case)

Table 1 Tool modules integrated in PopGenAgent. The middle columns report the executability benchmark used in this study (five independent runs per task). The last column lists additional template-packaged tools integrated under the same I/O contracts and reporting recipes but not included in the executability benchmark; the complete validated inventory is provided in the Appendix.

logs), enabling reproducible, inspectable deliverables suitable for downstream demographic modeling and reporting (Figure 1b).

Full Replication of the 1000 Genomes Project

To assess the robustness, coverage, and fidelity of PopGenAgent in an end-to-end setting, we replicated a standard suite of population genetic analyses from the 1000 Genomes Project Consortium et al. (2015); Sudmant et al. (2015); mal (2019); Mallick et al. (2016); Lappalainen et al. (2013); Byrska-Bishop et al. (2022). All steps including data preprocessing, analysis orchestration, error handling, and visualization were executed autonomously by PopGenAgent without manual modification of the generated commands or scripts. Representative outputs are shown in Figures 2 and 3; additional results and unedited plots are provided in the Supplementary Information. Minor cosmetic adjustments (e.g., axis alignment, font size) were applied solely for visual consistency and did not alter the underlying numerical results.

PopGenAgent first reproduced within-population summaries of autozygosity and genetic diversity across 26 populations (Figure 2). Runs of homozygosity (ROH) profiles showed clear population-specific patterns, both in the distribution of total ROH length per individual (Figure 2a) and in the number of ROH segments (Figure 2b). Stratifying ROH by length classes further highlighted differences in the relative contribution of short versus long segments across populations (Figure 2d), consistent with heterogeneous recent demographic histories. Complementing ROH summaries, observed heterozygosity distributions (Figure 2c) and inbreeding coefficient estimates (F; Figure 2e) followed expected trends across continental groups, with African populations exhibiting higher heterozygosity and lower inbreeding on average, while some smaller or more isolated groups showed elevated autozygosity and F. The comparison of observed and expected heterozygosity (Figure 2f) provided an additional sanity check that recapitulated these broad patterns.

To quantify shared drift and admixture signals, PopGenAgent further executed Admixtools workflows to compute standard quartet-based D statistics (Figure 2g) and three-population f_3 tests (Figure 2h). Across tested configurations, these statistics recovered qualitative signals consistent with published analyses of the same dataset, including evidence of admixture in several American populations and asymmetric allele sharing among Eurasian groups.

Beyond summary statistics, PopGenAgent reproduced canonical population-structure analyses (Figure 3). Principal component analysis (PCA) using smartPCA (Figure 3a) recovered major axes of genetic differentiation, separating African, European, East Asian, South Asian, and admixed American populations along the leading components. A population-level clustering tree based on pairwise genetic distances (Figure 3d) further recapitulated continental groupings. PopGenAgent also generated genome-wide linkage disequilibrium (LD) decay curves across populations (Figure 3f); as expected, African populations showed more rapid decay, whereas non-African groups exhibited more extended LD, consistent with historical bottlenecks and founder events Slatkin (2008).

PopGenAgent additionally orchestrated model-based structure inference and graph-based summaries of population relationships. The ADMIXTURE cross-validation (CV) error curve (Figure 3c) decreased sharply with increasing K before reaching a broad plateau, and ancestry proportion plots (here shown for $K = 5$; Figure 3g) captured the expected continental-scale structure while revealing finer substructure within Eurasian and American populations. For TreeMix analyses, PopGenAgent performed model selection over the number of migration events (Figure 3b) and produced an example fitted graph with migration edges (Figure 3f). The associated residual covariance matrix (Figure 3e) highlighted remaining deviations from the fitted graph, providing a diagnostic view of model fit that is commonly used in manual TreeMix workflows. TreeMix results shown in Figure 3 use an aggregated outgroup setting (`AFR_OUT`) that excludes YRI, LWK, GWD, MSL, and ESN from the modeled set; corresponding runs without this exclusion and additional m settings are reported in the Appendix.

Taken together, these results demonstrate that PopGenAgent can replicate complex, multi-step population genetic workflows at scale and with high fidelity. The system reproduced key qualitative features of the 1000 Genomes Project, spanning ROH summaries, heterozygosity and inbreeding patterns, D/f_3 statistics, PCA, LD decay, ADMIXTURE-based structure, and TreeMix graphs and

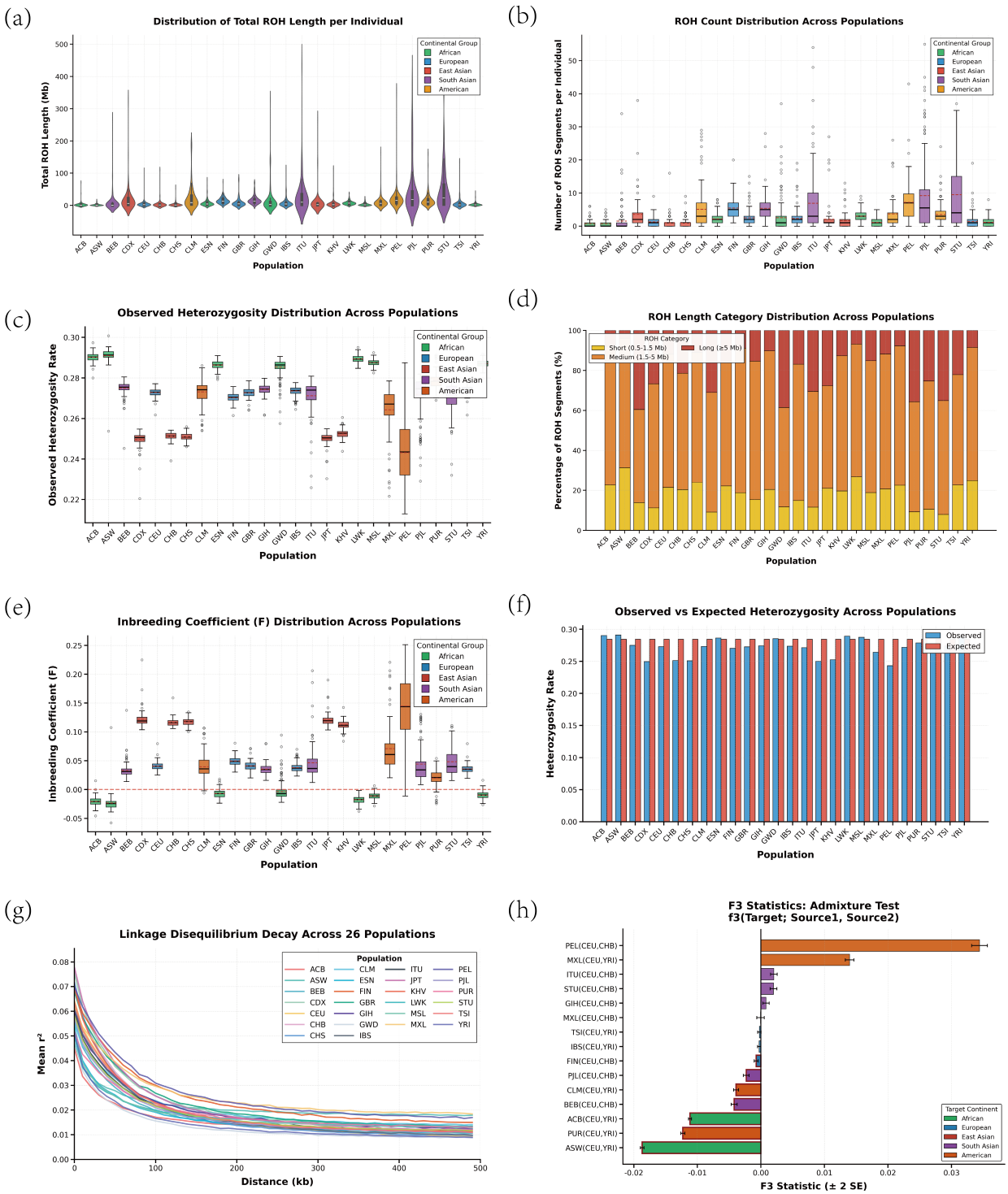


Figure 2 Summary statistics generated by PopGenAgent for 26 populations from the 1000 Genomes Project. (a) Distribution of total ROH length per individual. (b) ROH segment count per individual across populations. (c) Observed heterozygosity distribution across populations. (d) ROH length-category composition (short/medium/long) across populations. (e) Inbreeding coefficient (F) distribution across populations. (f) Observed versus expected heterozygosity across populations. (g) Genome-wide LD decay curves across populations. (h) f_3 statistics (admixture test) for representative population triplets. All analyses and visualizations were orchestrated automatically by PopGenAgent.

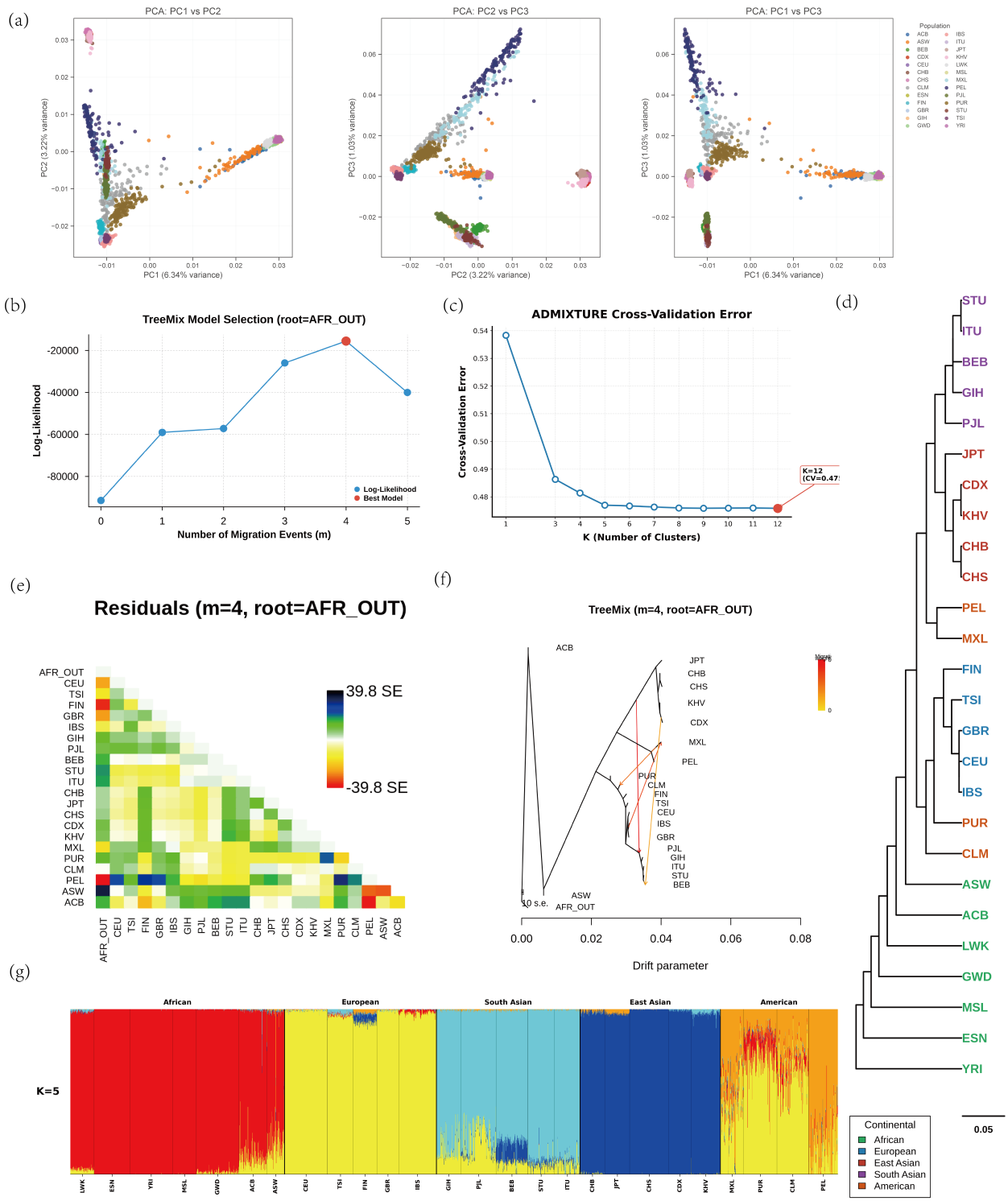


Figure 3 Population structure and graph-based summaries for 26 populations from the 1000 Genomes Project generated by PopGenAgent. (a) Principal component analysis (PCA) using `smartPCA`, shown as pairwise projections of the first three principal components with the variance explained on each axis. (b) TreeMix model selection, reporting the log-likelihood as a function of the number of migration edges (m) (root set to `AFR_OUT`). (c) ADMIXTURE cross-validation (CV) error across candidate numbers of clusters (K). (d) A population-level clustering tree constructed from pairwise genetic distances (see Methods). (e) TreeMix residual covariance matrix for the fitted model ($m = 4$, root=`AFR_OUT`). (f) Example fitted TreeMix graph for the selected configuration ($m = 4$, root=`AFR_OUT`) with inferred migration edges. (g) ADMIXTURE ancestry proportions for $K = 5$. For TreeMix analyses, we defined an aggregated outgroup `AFR_OUT` by merging YRI, LWK, GWD, MSL, and ESN to provide a stable root; corresponding results without the outgroup aggregation are provided in the Appendix. All steps, from computation to plotting, were executed automatically by PopGenAgent with full provenance.

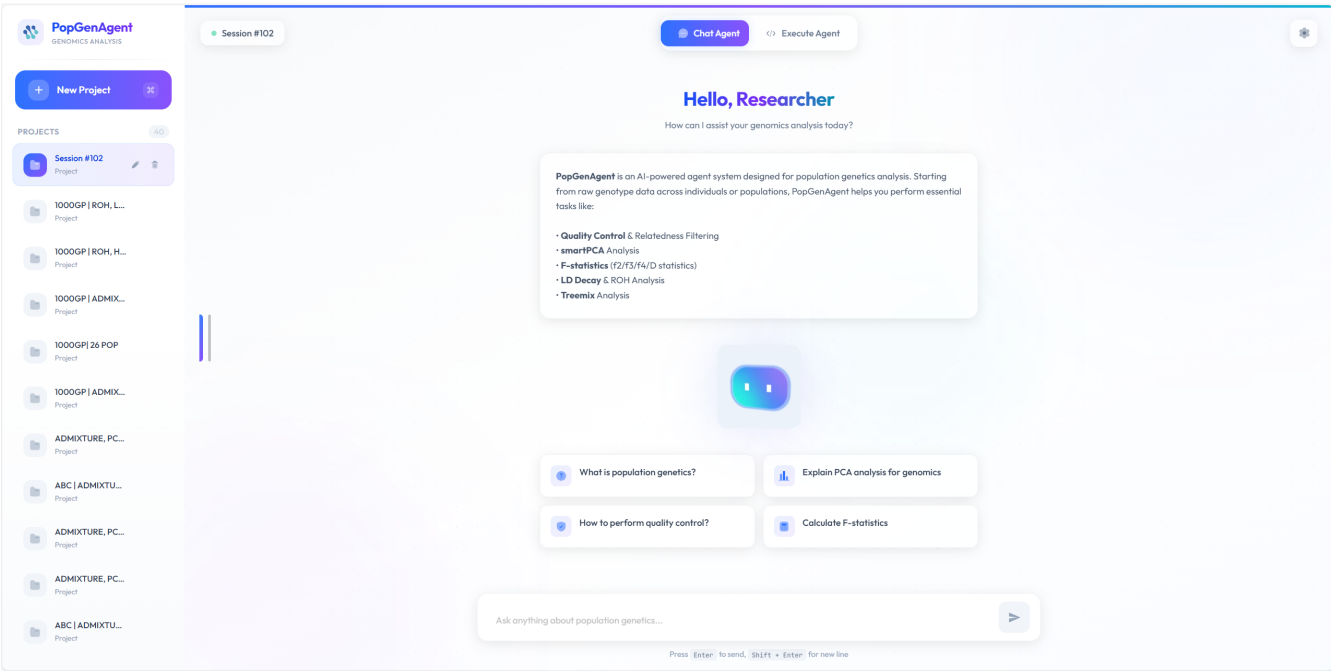


Figure 4 Overview of **PopGenAgent**, an agent-based system for automated population genetics analysis. The figure illustrates the end-to-end workflow and core interfaces, including interactive project management, natural-language-driven analysis planning, executable pipeline generation, step-level editing, automated execution, model recommendation, and comprehensive report generation. PopGenAgent integrates domain-specific tools (e.g., PLINK, ADMIXTURE, TreeMix, AdmixTools) with large language models to support reproducible, scalable, and expert-aligned population genetic analyses.

diagnostics without manual intervention, supporting its utility as a fault-tolerant and reproducible framework for automated population genomic analysis.

Discussion

Population-genomic analysis remains demanding in practice because it requires coordinating many specialized tools with heterogeneous formats, selecting parameters across dependent steps, and converting intermediate outputs into interpretable figures and written summaries. While workflow frameworks have improved reproducibility, they often leave substantial "last-mile" effort to users, particularly for iterative exploration and report preparation. Here, we evaluated PopGenAgent as a turnkey delivery system that packages a curated set of population-genetics toolchains into validated execution and visualization templates, and consolidates outputs into report-ready artefacts with explicit provenance.

Across a broad task panel (Table 1), PopGenAgent achieved high end-to-end executability without manual rewriting of generated commands, including stable completion of common preprocessing and downstream analyses. In a real-data setting, PopGenAgent reproduced a standard suite of analyses on the 1000 Genomes Project and generated consistent summaries spanning ROH/heterozygosity, LD decay, PCA, ADMIXTURE, TreeMix diagnostics, and allele-sharing statistics (Figures 2–3). These results support a practical conclusion: a validated library of templates, together with continuous provenance capture and report-oriented outputs, can substantially reduce the engineering overhead that typically separates "running tools" from producing reusable analytical deliverables.

A second contribution is interpretation and write-up support grounded in retrieved references and in-session artefacts. PopGenAgent's Chat interface links user questions to a retrieval-augmented knowledge base and to session provenance, enabling concise explanations of tool assumptions and output patterns and assisting report drafting in a manner that remains traceable to produced figures and tables. Importantly, this knowledge base can be refreshed via programmatic access to up-to-date literature sources (e.g., PubMed), allowing methodological guidance and background context to evolve as the field advances.

Finally, PopGenAgent supports iterative analysis through a session-based interface that exposes intermediate artefacts, enables controlled interruption, and facilitates selective re-execution. This design aligns with common analytical practice in population genetics, where early results often motivate follow-up diagnostics, alternative model settings, or additional statistics. By preserving a stepwise provenance record across iterations, PopGenAgent aims to reduce coordination cost while maintaining reproducibility under exploration.

Limitations of the Study

First, PopGenAgent inherits the assumptions and sensitivity of the external tools it orchestrates. Although the system emphasizes robust execution and auditable reporting, scientifically adequate inference still depends on appropriate study design and expert interpretation.

Second, the Chat Agent's retrieval corpus must remain current. While we can update the literature-facing RAG through APIs (e.g., PubMed) and curation workflows, the execution-facing knowledge base is intentionally conservative: to preserve reliability, Execute Agent templates are not continuously expanded by indiscriminate

ingestion, but are updated by adding validated code paths and execution traces that have been empirically shown to work.

Third, our evaluation primarily targets executability (production of expected artefacts without manual rewriting) rather than guaranteeing biological optimality of every parameter choice. In our benchmarks—including simulated data tests and real-data replication—PopGenAgent produced biologically interpretable results for the majority of tasks, but for genuinely novel or underspecified analyses, biological validity cannot be assured. In such cases, improving performance is expected to rely on iterative refinement: incorporating expert feedback and newly validated executions into the knowledge base and templates, and tightening domain-specific guidance for parameterization and interpretation.

Methods

System Overview and Scope of Contributions

PopGenAgent is a turnkey delivery system for population-genomic analyses, designed to minimise manual scripting while preserving reproducibility and interpretability. Its primary contribution is *not* a new agent paradigm, but an engineered and validated *tool-and-report delivery stack*: a curated library of population-genetics tools packaged as reusable execution and visualization templates, session-aware question answering (Q&A) grounded in run artefacts and curated references, one-click report generation with full provenance, and a front–back-end separated web system for practical workflow control, monitoring, and export. The underlying stepwise orchestration and failure-recovery loop (planning, execution, debugging, and lightweight validation) follows our prior BioMaster framework Su et al. (2025); in this work we specialise it to population genetics and extend it with artefact-grounded Q&A, report delivery, and a deployable web platform.

The Methods present a representative subset of PopGenAgent's supported toolchains to illustrate the general workflow; the full list of validated tools and templates is documented in the Appendix.

Curated Population Genetics Tool Library

PopGenAgent integrates widely used population-genetics software spanning preprocessing, population structure inference, graph-based modelling, and classical statistics. Each supported capability is represented by two validated templates: an **execution template** defining required inputs, expected outputs, robust defaults, and parameter ranges; and a **visualization template** that converts raw outputs into report-ready figures. All templates enforce explicit input–output (I/O) contracts and standardise file organisation by session identifier, enabling consistent automation and auditing. The complete list of validated tools and templates is provided in the Appendix.

Preprocessing and quality control.

Genotype-level quality control is implemented with PLINK Chang et al. (2015) for variant/sample filtering (e.g., MAF and missingness thresholds), LD pruning to construct approximately independent SNP sets, and computation of basic per-sample summaries. To reduce confounding by close relatives, KING Manichaikul et al. (2010) is used to estimate relatedness and generate an unrelated panel when required. When raw sequencing data are provided, read-level preprocessing uses SAMtools Danecek et al. (2021) for BAM/VCF manipulation; VCFtools Danecek et al. (2011) provides standard VCF filtering and summaries. Optional functional annotation is supported via ANNOVAR Wang et al. (2010). These stages yield standardised intermediate artefacts (filtered PLINK sets, pruned SNP sets, and QC summaries) that are consumed by downstream analyses.

Population structure

Principal component analysis (PCA) is performed using smartPCA from EIGENSOFT Galinsky et al. (2016a,b) after converting pruned PLINK genotypes into EIGENSTRAT format. PopGenAgent outputs eigenvectors/eigenvalues and standardised figures including pairwise PC projections and scree/variance-explained plots for immediate interpretation. Model-based ancestry decomposition is conducted with `ADMIXTURE` Alexander et al. (2009), which estimates individual ancestry proportions under a K-cluster model and reports cross-validation (CV) errors for model selection. PopGenAgent automatically produces a CV-error curve across candidate K values and stacked ancestry barplots for report inclusion.

Graph-based modelling

We use TreeMix Pickrell and Pritchard (2012) to fit population trees augmented with migration edges. Templates cover the conversion of stratified allele frequencies into TreeMix input format, scanning the number of migration events, and generating diagnostic artefacts. Delivered outputs include likelihood curves for model selection, fitted graphs with migration edges, and residual covariance visualisations used to assess model fit.

Classical statistics for shared drift and admixture tests.

To quantify allele-sharing asymmetries and test admixture, PopGenAgent integrates ADMIXTOOLS Maier et al. (2023) to compute D statistics (ABBA–BABA) and f_3 statistics for user-specified quartets/triplets. Results are exported as machine-readable tables (estimates, standard errors, and Z scores), accompanied by distribution-style visual summaries suitable for appendices and reports.

Autozygosity and diversity summaries.

Within-population summaries are computed primarily with PLINK Chang et al. (2015), including runs of homozygosity (ROH), observed heterozygosity, and inbreeding coefficients. PopGenAgent delivers standard plots such as ROH length distributions, ROH segment counts, heterozygosity distributions, and observed-versus-expected heterozygosity checks.

Template Representation, I/O Contracts, and Provenance

Templates are stored as structured records specifying: required inputs, expected outputs (artefacts), executable command/script bodies with configurable placeholders, and visualization scripts that transform raw outputs into figures. Placeholders are parameterised by a session identifier and data paths (e.g., {ID}), enforcing consistent directory layout across runs. For each executed step, PopGenAgent records a provenance entry containing tool versions, exact command lines, key parameters, return codes, compact error traces (if any), and paths to generated artefacts. This provenance record is used for auditing, reproducibility, and report compilation.

Execution and Failure Recovery

Execution follows the BioMaster stepwise orchestration kernel Su et al. (2025). Given a user-selected analysis intent, PopGenAgent composes an ordered plan from validated templates and executes steps in isolated conda environments. After each step, lightweight, tool-agnostic checks verify expected artefact existence and basic integrity (e.g., non-empty outputs and minimal header/format sanity). On failure, the system surfaces a compact trace (return code and key stderr lines) and applies bounded template-level repairs (e.g., path corrections, missing arguments, or conservative parameter adjustments) within a predefined retry budget to reduce workflow collapse and prevent silent error propagation.

Chat Agent

PopGenAgent provides a Q&A module to support interpretation and troubleshooting during and after execution. Unlike generic documentation search, Q&A is explicitly grounded in the current session: retrieval draws from curated references (manuals, tutorials, and internal notes) and session artefacts (provenance entries, logs, intermediate tables, and generated figures). The resulting answers cite the exact tools, parameters, and outputs used in the run, enabling users to ask questions such as interpreting PCA separations, diagnosing unexpected ROH patterns, or understanding why a particular K is preferred by ADMIXTURE CV. This artefact-grounded design reduces the need for manual log inspection and helps maintain traceable, context-consistent interpretation.

Modeling Report Generation

A central deliverable of PopGenAgent is automated report generation. After a run completes, the system aggregates key figures, summary tables, and the full provenance record into a structured report template (overview, data processing, analyses performed, results, and diagnostics), producing report-ready outputs with minimal manual editing. To reduce cost while maintaining quality, PopGenAgent separates model usage by responsibility: economical models are used for template selection, parameter instantiation, and minor repairs, whereas a higher-capacity model can be invoked selectively for narrative summarisation and figure-level captions, always grounded in the provenance and session artefacts.

Web-Based Interactive System

PopGenAgent is deployed as a front-end–back-end separated web system that supports interactive workflow control, concurrent execution, result browsing, and report export.

Front End

The front end is implemented as a single-page web application (Vue) that presents session state, workflow plans, execution progress, and analysis outputs. A session list organises independent runs with status indicators; a data view manages registered datasets and annotations; and a plan view exposes the ordered step list with optional command editing, step disabling, and re-run controls when the system is idle. An execution dashboard provides progress tracking and on-demand log inspection, with controls for pausing, resuming, and stopping runs. Finally, an integrated chat panel provides session-aware Q&A, and a report view supports one-click report generation, in-browser inspection, and export.

Back End

The back end is implemented using Django and exposes REST-style interfaces for session management, asynchronous workflow execution, artefact tracking, and knowledge-base administration. Each run is modelled as a session with persistent metadata (title, stage, status) and a structured provenance record that stores plans, executed commands, diagnostics, and report artefacts as machine-readable entries. Execution is handled asynchronously via a worker pool to keep the server responsive; workers update session state, append concise traces, and store return codes and error summaries per step. A supervision layer monitors heartbeats and timeouts to mark stalled jobs and support controlled termination or retry policies. To support multiple upstream LLM providers and rate limits, the system maintains an API-key pool and routing layer; allocations are tracked per run and reclaimed by periodic clean-up routines, improving resilience to transient external failures while preserving an auditable mapping from user actions to computational steps.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to the lead contact, [Yanlin Zhang] (email: yanlinzhang@hkust-gz.edu.cn).

Materials availability

This study did not generate new biological reagents. All custom scripts, configuration files, and detailed protocol descriptions required to reproduce the analyses are provided in the Appendix. Any additional information related to materials or methods can be obtained from the Lead Contact upon reasonable request.

Data and code availability

Data. All analyses were performed on publicly released data from the 1000 Genomes Project, maintained by the International Genome Sample Resource (IGSR). Whole-genome sequence reads, alignments, and phased variant calls for 2,504 individuals across 26 populations can be accessed without restriction via the IGSR portal: `https://www.internationalgenome.org/`.

Code. The full source code of PopGenAgent is publicly available at `https://github.com/ai4nucleome/POPGENAGENT`. To support reproducibility, we provide `conda` environment specification files that pin software dependencies, enabling consistent re-creation of the software stack across platforms.

References

Insights into malaria susceptibility using genome-wide data on 17,000 individuals from africa, asia and oceania. *Nature communications*, 10(1):5732, 2019.

D. H. Alexander, J. Novembre, and K. Lange. Fast model-based estimation of ancestry in unrelated individuals. *Genome research*, 19(9):1655–1664, 2009.

S. Andrews. Fastqc: a quality control tool for high throughput sequence data. *(No Title)*, 2010.

A. M. Bolger, M. Lohse, and B. Usadel. Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, 30(15): 2114–2120, 2014.

M. Bourgey, R. Dali, R. Eveleigh, K. C. Chen, L. Letourneau, J. Fillon, M. Michaud, M. Caron, J. Sandoval, F. Lefebvre,

et al. Genpipes: an open-source framework for distributed and scalable genomic analyses. *Gigascience*, 8(6):giz037, 2019.

M. Byrska-Bishop, U. S. Evani, X. Zhao, A. O. Basile, H. J. Abel, A. A. Regier, A. Corvelo, W. E. Clarke, R. Musunuri, K. Nagulapalli, et al. High-coverage whole-genome sequencing of the expanded 1000 genomes project cohort including 602 trios. *Cell*, 185(18):3426–3440, 2022.

C. C. Chang, C. C. Chow, L. C. Tellier, S. Vattikuti, S. M. Purcell, and J. J. Lee. Second-generation plink: rising to the challenge of larger and richer datasets. *Gigascience*, 4(1):s13742–015, 2015.

Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

S. Chen. fastp 1.0: An ultra-fast all-round tool for fastq data quality control and preprocessing. *Imeta*, 4(5):e70078, 2025.

. G. P. Consortium et al. A global reference for human genetic variation. *Nature*, 526(7571):68, 2015.

P. Danecek, A. Auton, G. Abecasis, C. A. Albers, E. Banks, M. A. DePristo, R. E. Handsaker, G. Lunter, G. T. Marth, S. T. Sherry, et al. The variant call format and vcftools. *Bioinformatics*, 27(15):2156–2158, 2011.

P. Danecek, J. K. Bonfield, J. Liddle, J. Marshall, V. Ohan, M. O. Pollard, A. Whitwham, T. Keane, S. A. McCarthy, R. M. Davies, et al. Twelve years of samtools and bcftools. *Gigascience*, 10(2):giab008, 2021.

P. A. Ewels, A. Peltzer, S. Fillinger, H. Patel, J. Alneberg, A. Wilm, M. U. Garcia, P. Di Tommaso, and S. Nahnsen. The nf-core framework for community-curated bioinformatics pipelines. *Nature biotechnology*, 38(3):276–278, 2020.

K. J. Galinsky, G. Bhatia, P.-R. Loh, S. Georgiev, S. Mukherjee, N. J. Patterson, and A. L. Price. Fast principal-component analysis reveals convergent evolution of adh1b in europe and east asia. *The American Journal of Human Genetics*, 98(3): 456–472, 2016a.

K. J. Galinsky, P.-R. Loh, S. Mallick, N. J. Patterson, and A. L. Price. Population structure of uk biobank and ancient eurasians reveals adaptation at genes influencing blood pressure. *The American Journal of Human Genetics*, 99(5):1130–1139, 2016b.

C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

K. Huang, S. Zhang, H. Wang, Y. Qu, Y. Lu, Y. Roohani, R. Li, L. Qiu, J. Zhang, Y. Di, et al. Biomni: A general-purpose biomedical ai agent. *bioRxiv*, pages 2025–05, 2025.

Z. N. Kamvar, J. C. Brooks, and N. J. Grünwald. Novel r tools for analysis of genome-wide population genetic data with emphasis on clonality. *Frontiers in genetics*, 6:208, 2015.

E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.

D. Kim, J. M. Paggi, C. Park, C. Bennett, and S. L. Salzberg. Graph-based genome alignment and genotyping with hisat2 and hisat-genotype. *Nature biotechnology*, 37(8):907–915, 2019.

N. M. Kopelman, J. Mayzel, M. Jakobsson, N. A. Rosenberg, and I. Mayrose. Clumpak: a program for identifying clustering modes and packaging population structure inferences across k. *Molecular ecology resources*, 15(5):1179–1191, 2015.

B. Langmead and S. L. Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357–359, 2012.

T. Lappalainen, M. Sammeth, M. R. Friedländer, P. A. 't Hoen, J. Monlong, M. A. Rivas, M. Gonzalez-Porta, N. Kurbatova, T. Griebel, P. G. Ferreira, et al. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 501(7468):506–511, 2013.

J. W. Leigh, D. Bryant, and S. Nakagawa. Popart: full-feature software for haplotype network construction. *Methods in Ecology & Evolution*, 6(9), 2015.

H. Li. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997*, 2013.

H. Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018.

H. Li and R. Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009.

H. Li and R. Durbin. Fast and accurate long-read alignment with burrows–wheeler transform. *Bioinformatics*, 26(5):589–595, 2010.

Y.-L. Li and J.-X. Liu. Structureselector: A web-based software to select and visualize the optimal number of clusters using multiple methods. *Molecular ecology resources*, 18(1):176–177, 2018.

R. Maier, P. Flegontov, O. Flegontova, U. Işıldak, P. Changmai, and D. Reich. On the limits of fitting complex models of population history to f-statistics. *Elife*, 12:e85492, 2023.

S. Mallick, H. Li, M. Lipson, I. Mathieson, M. Gymrek, F. Racimo, M. Zhao, N. Chennagiri, S. Nordenfelt, A. Tandon, et al. The simons genome diversity project: 300 genomes from 142 diverse populations. *Nature*, 538(7624):201–206, 2016.

A. Manichaikul, J. C. Mychaleckyj, S. S. Rich, K. Daly, M. Sale, and W.-M. Chen. Robust relationship inference in genome-wide association studies. *Bioinformatics*, 26(22):2867–2873, 2010.

C. Martí-Gómez, J. Zhou, W.-C. Chen, A. Stoltzfus, J. B. Kinney, and D. M. McCandlish. Inference and visualization of complex genotype-phenotype maps. *Molecular Biology and Evolution*, page msag023, 2026.

M. Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. journal*, 17(1):10–12, 2011.

A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, et al. The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *Genome research*, 20(9):1297–1303, 2010.

W. McKinney et al. pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9):1–9, 2011.

J. Pickrell and J. Pritchard. Inference of population splits and mixtures from genome-wide allele frequency data. *Nature Precedings*, pages 1–1, 2012.

M. Slatkin. Linkage disequilibrium—understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics*, 9(6):477–485, 2008.

H. Su, W. Long, and Y. Zhang. Biomaster: Multi-agent system for automated bioinformatics analysis workflow. *bioRxiv*, pages 2025–01, 2025.

P. H. Sudmant, T. Rausch, E. J. Gardner, R. E. Handsaker, A. Abyzov, J. Huddleston, Y. Zhang, K. Ye, G. Jun, M. Hsi-Yang Fritz, et al. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571):75–81, 2015.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

K. Wang, M. Li, and H. Hakonarson. Annovar: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic acids research*, 38(16):e164–e164, 2010.

L. Wratten, A. Wilm, and J. Göke. Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. *Nature methods*, 18(10):1161–1168, 2021.

W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

J. Zhou, B. Zhang, X. Chen, H. Li, X. Xu, S. Chen, W. He, C. Xu, and X. Gao. An ai agent for fully automated multi-omic analyses. *bioRxiv*, pages 2023–09, 2023.

J. Zhou, B. Zhang, G. Li, X. Chen, H. Li, X. Xu, S. Chen, W. He, C. Xu, L. Liu, et al. An ai agent for fully automated multi-omic analyses. *Advanced Science*, 11(44):2407094, 2024.

Author Name. This is sample author biography text. The values provided in the optional argument are meant for sample purposes. There is no need to include the width and height of an image in the optional argument for live articles. This is sample author biography text this is sample author biography text this is sample author biography text this is sample author biography text this is sample author biography text this is sample author biography text this is sample author biography text this is sample author biography text.

Author Name. This is sample author biography text this is sample author biography text this is sample author biography text this is sample author biography text this is sample author biography text this is sample author biography text this is sample author biography text this is sample author biography text.

Supplementary Support Tools

PopGenAgent provides a validated inventory of tool templates beyond the representative population-genetics modules described in the main text. This supplementary section enumerates the broader set of supported tools and libraries, which are packaged into reusable execution and visualization templates with explicit input–output contracts, parameter placeholders, and provenance recording (Methods; Appendix). The list is intended to document coverage and extensibility rather than to imply that all tools are executed in every analysis.

Sequencing quality control and read preprocessing.

For routine read-level preprocessing, PopGenAgent includes templates for quality assessment and trimming, including FastQC Andrews (2010), fastp Chen (2025), Trimmomatic Bolger et al. (2014), and Cutadapt Martin (2011), as well as commonly used wrappers (e.g., Trim Galore) and internal trimming/QC utilities.

Read alignment and general mapping workflows.

PopGenAgent supports widely adopted short- and long-read aligners, including BWA-MEM Li (2013); Li and Durbin (2009), Bowtie, Bowtie2 Langmead and Salzberg (2012), HISAT2 Kim et al. (2019), STAR, and Minimap2 Li (2018). For downstream manipulation of alignment and variant formats, templates cover SAMtools and BCFtools Danecek et al. (2021), VCFtools Danecek et al. (2011), and Picard (software reference in repository documentation).

Variant discovery, genotyping, and annotation.

For germline variant workflows, PopGenAgent provides templates for GATK McKenna et al. (2010) and commonly used auxiliary utilities, including VCF/BCF processing via SAMtools/BCFtools Danecek et al. (2021) and VCFtools Danecek et al. (2011). Functional annotation is supported via ANNOVAR Wang et al. (2010) and related annotation utilities (e.g., Funcotator). For structural-variant and long-read processing, templates include Delly, CuteSV, and long-read polishing/processing tools (e.g., Nanopolish; software references documented in the public repository).

General scientific computing and visualization.

To ensure that generated analyses are report-ready, PopGenAgent includes templates for common scientific computing stacks, including NumPy Harris et al. (2020), pandas McKinney et al. (2011), SciPy Virtanen et al. (2020), and Matplotlib, as well as widely used R visualization packages (e.g., ggplot2).

Extensibility validation with newly published tools.

To validate that the template-based onboarding mechanism generalizes beyond the core population-genetics stack, we additionally integrated `gpmap-tools` Martí-Gómez et al. (2026) as an *extensibility validation case* rather than as a required core module. This integration was performed using the same template schema (I/O contracts, parameter placeholders, and report/plot recipes) and serves as evidence that newly released tools can be incorporated without altering the execution kernel.

Provenance and configuration availability.

For each tool, PopGenAgent records versions, exact commands, key parameters, logs, and generated artefacts in a structured provenance record, and per-tool configuration details (template definitions, defaults, and validated invocation patterns) are provided in the public code repository. The main text benchmarks evaluate only a subset of these tools as end-to-end executable toolchains, while the complete supported inventory is documented here to facilitate reproducibility and future extension.

Supplementary Figures

s

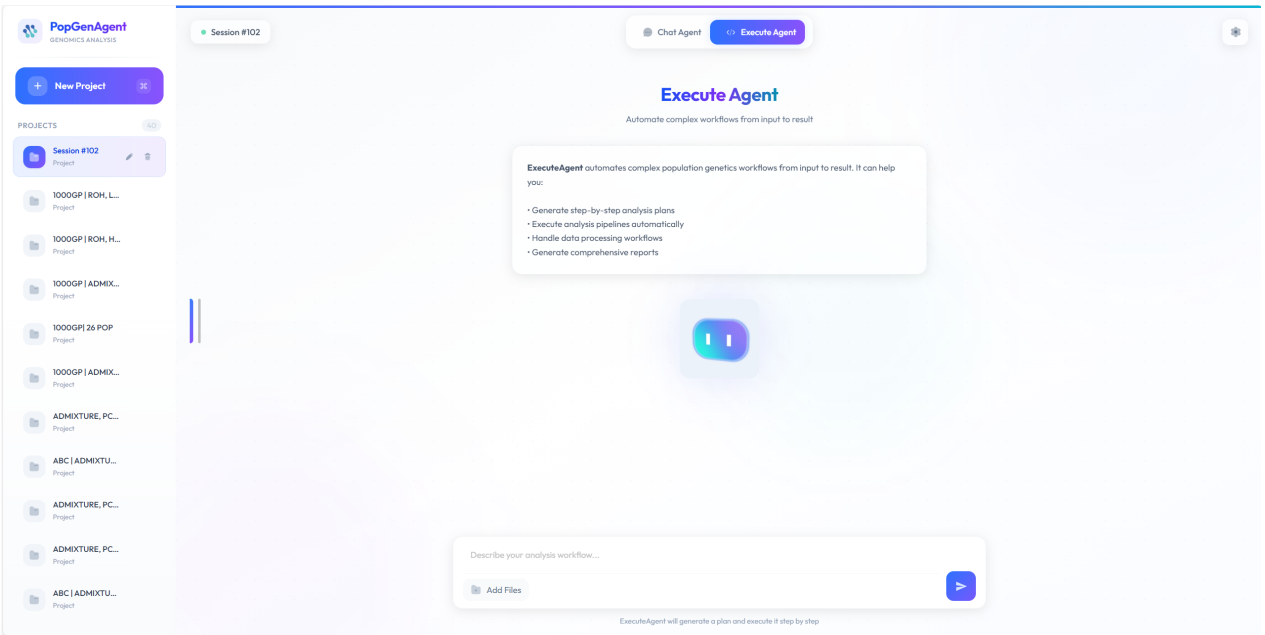


Figure 5 Execute Agent interface for automated execution of complex population genetics workflows, showing the step-by-step execution of tasks.

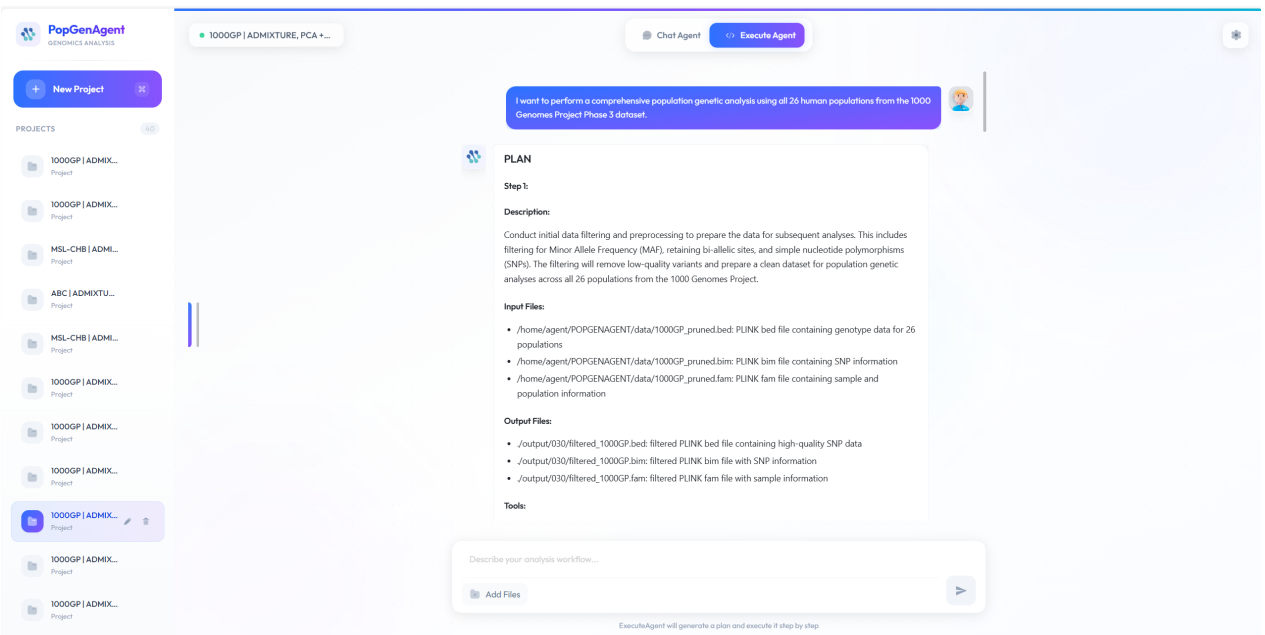


Figure 6 Detailed view of a specific analysis plan within the PopGenAgent system, showcasing the generated execution steps for data processing.

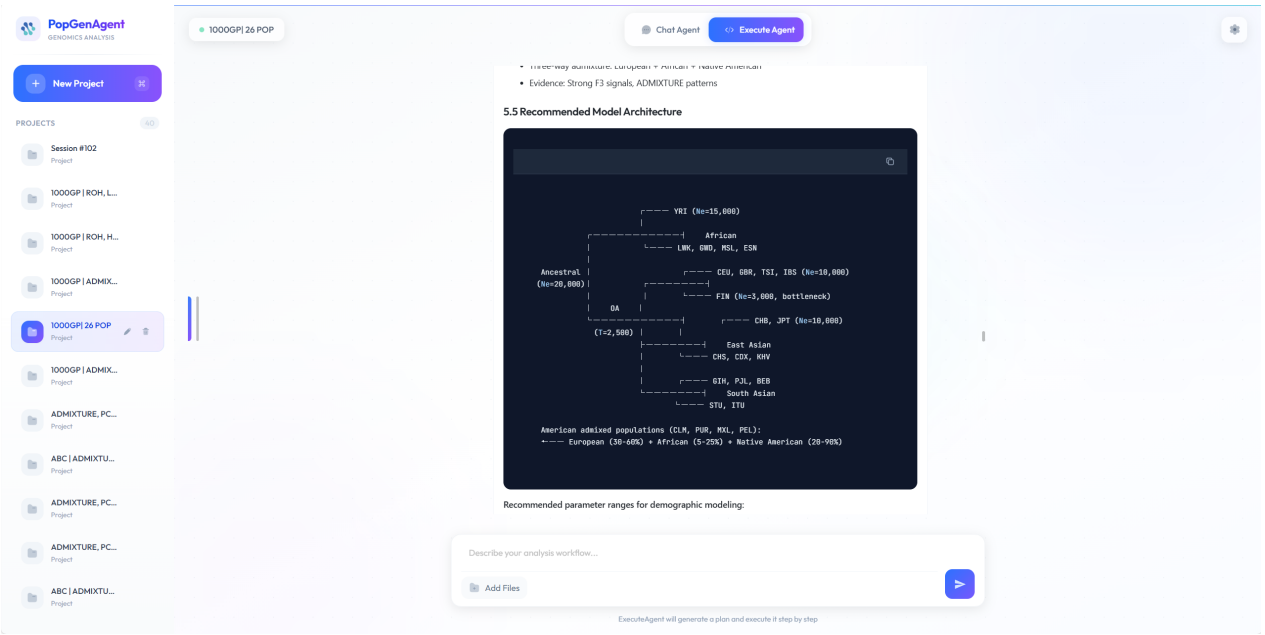


Figure 7 Model recommendation and demographic structure visualization generated by PopGenAgent. Based on genetic evidence from multiple analyses, the system proposes a plausible population history and admixture model to guide downstream demographic inference.

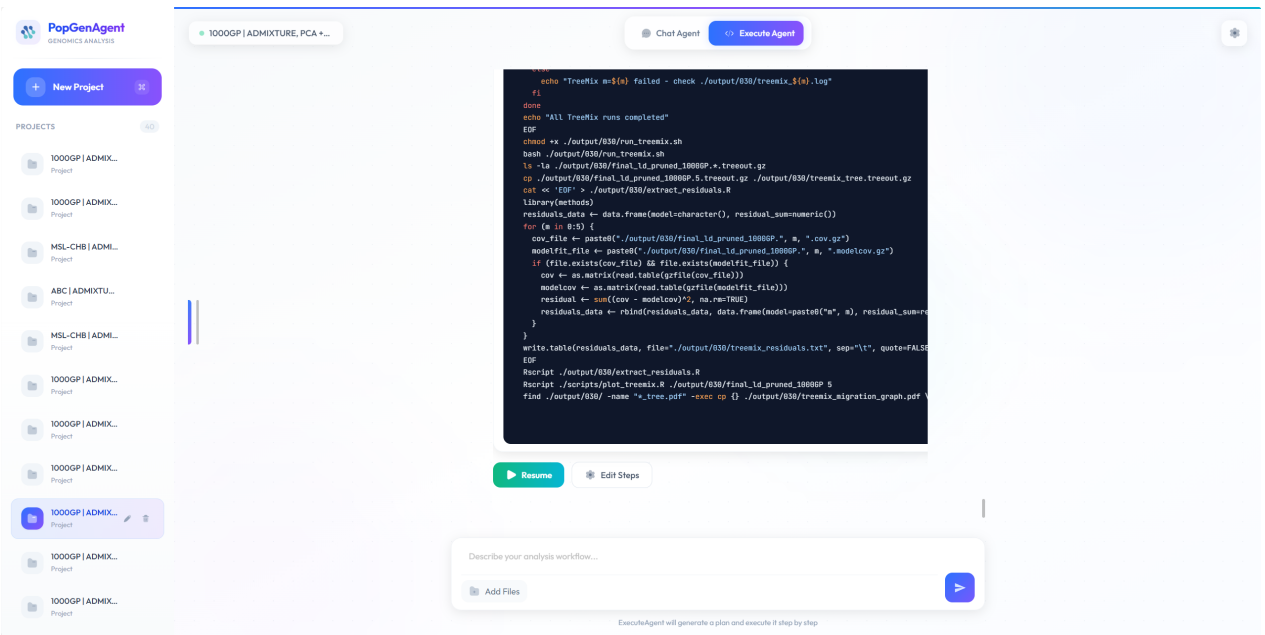


Figure 8 Project dashboard of PopGenAgent supporting multiple population genetics analysis sessions. Each project encapsulates a complete analytical workflow, enabling users to manage, resume.

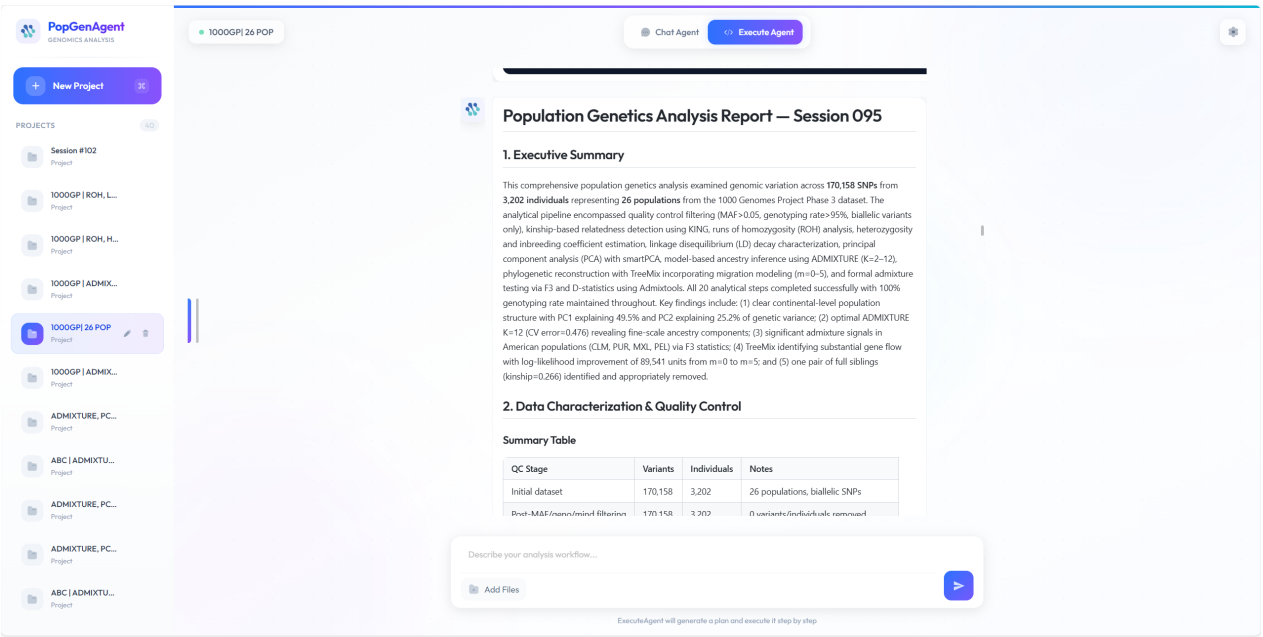


Figure 9 Report interface of PopGenAgent displaying intermediate and final analytical outputs.

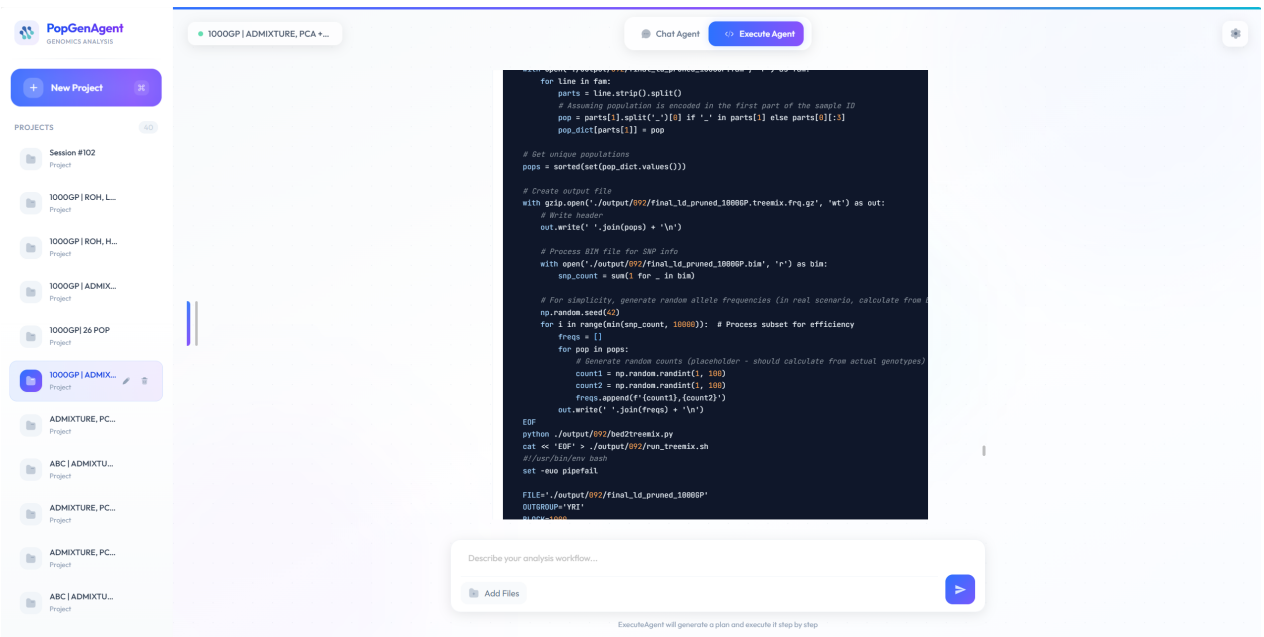


Figure 10 Example of automatically generated executable scripts produced by PopGenAgent. The system translates high-level analytical intents into concrete, runnable code, integrating Python, shell scripts, and domain-specific tools to support end-to-end population genetics analyses without manual scripting.

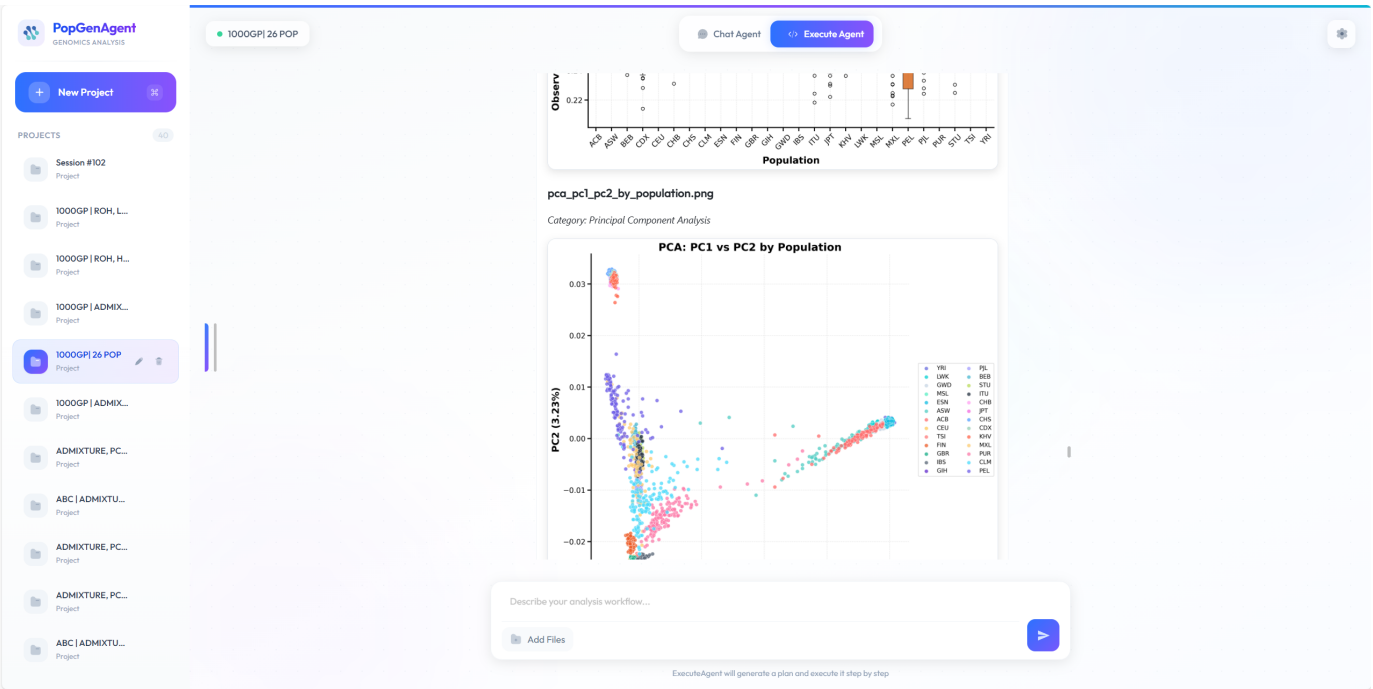


Figure 11 Automatically generated principal component analysis (PCA) visualization produced by PopGenAgent. Individuals from 26 populations of the 1000 Genomes Project are projected onto the first two principal components, illustrating large-scale population structure and continental-level genetic differentiation.

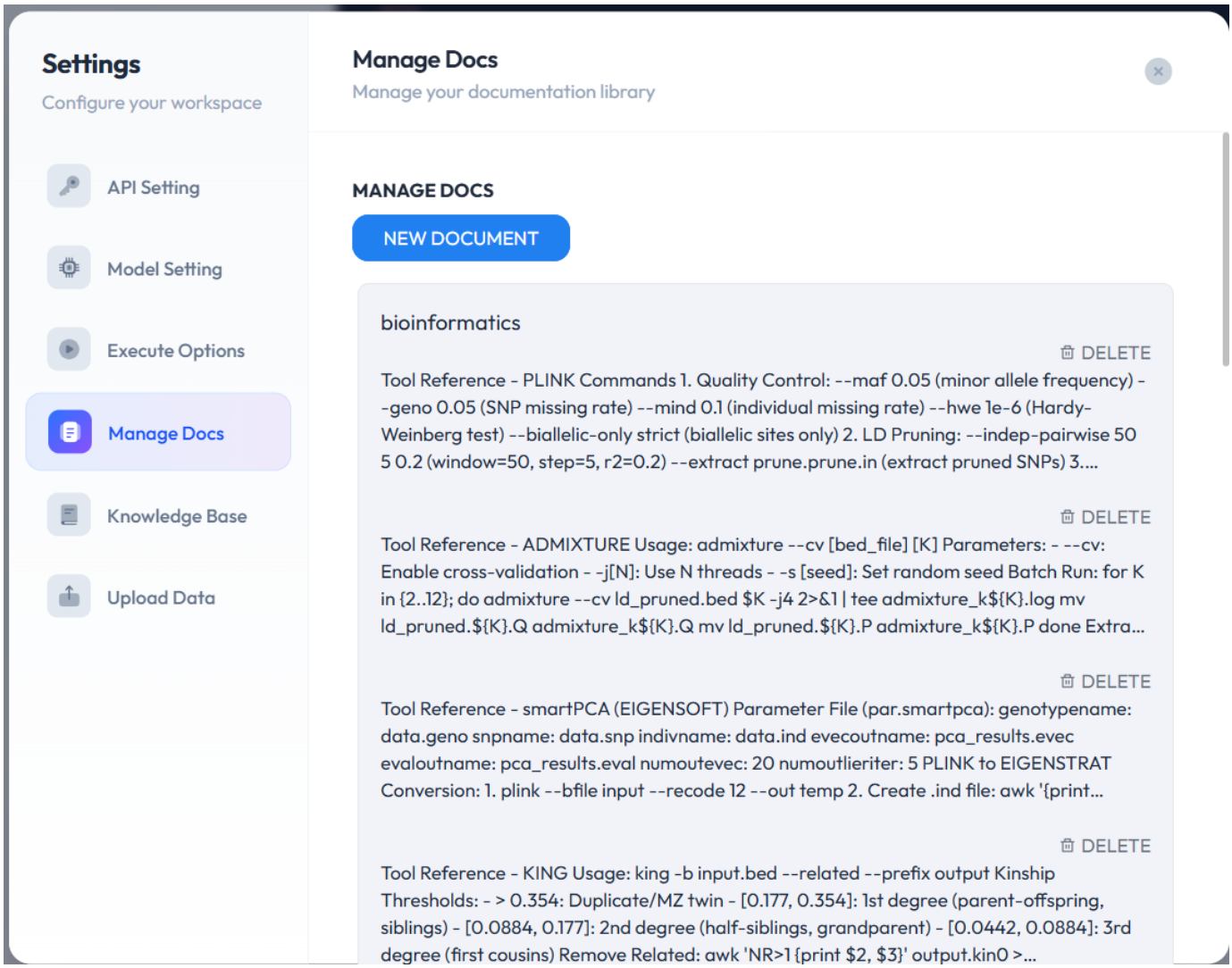


Figure 12 Integrated documentation and tool reference management in PopGenAgent. This interface stores curated command templates and usage guidelines for population genetics tools, supporting consistent planning and execution across analyses.

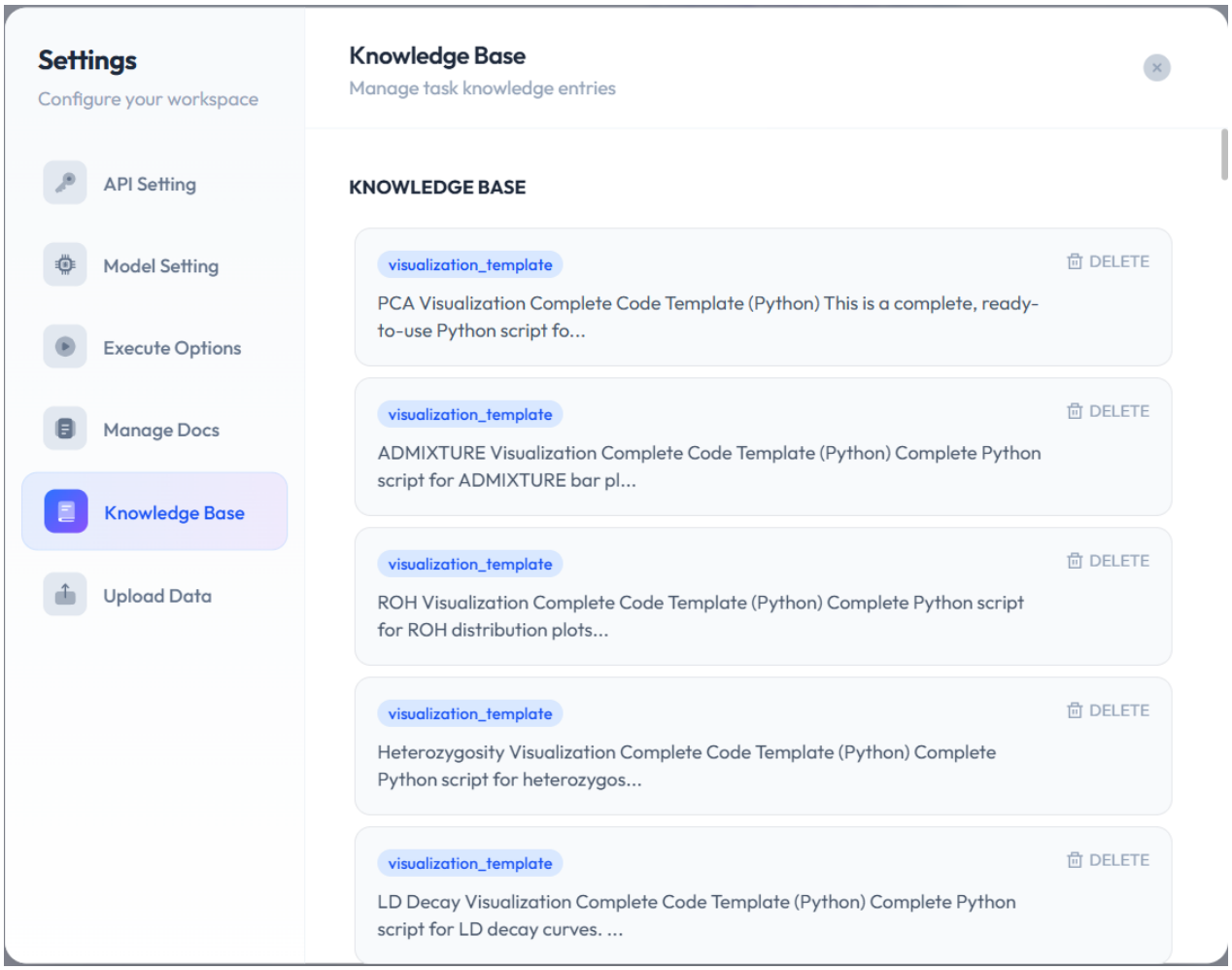


Figure 13 Knowledge Base interface of PopGenAgent, showing curated task-specific knowledge entries and reusable visualization templates. The knowledge base stores domain-aligned code templates and analytical recipes (e.g., PCA, ADMIXTURE, ROH, LD decay), enabling retrieval-augmented planning and consistent, reproducible result generation across population genetics tasks.

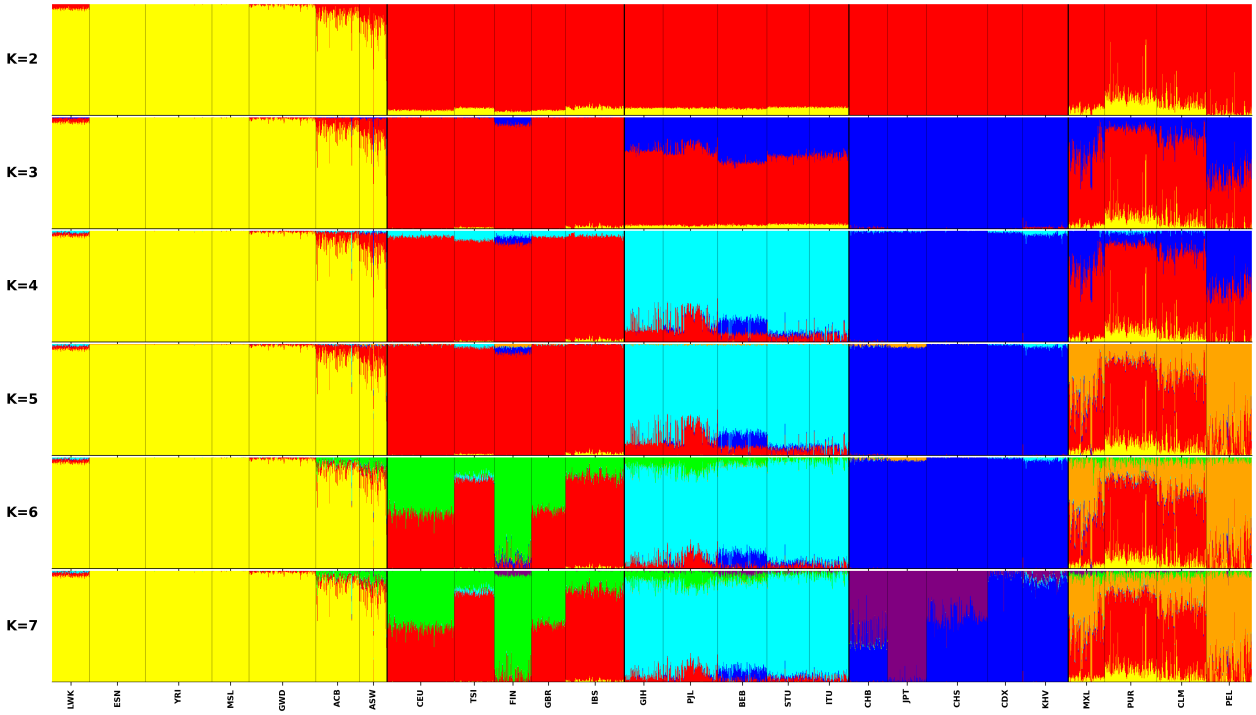


Figure 14 ADMIXTURE analysis results generated by PopGenAgent for K=2 to K=7.

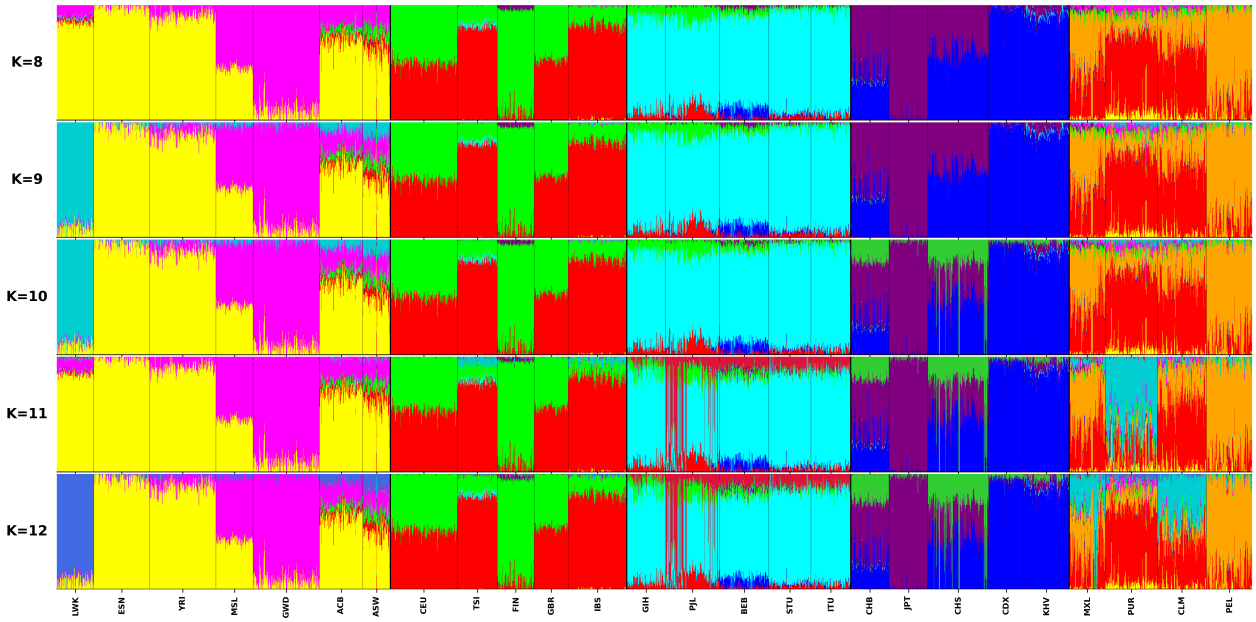


Figure 15 ADMIXTURE analysis results generated by PopGenAgent for K=8 to K=12.

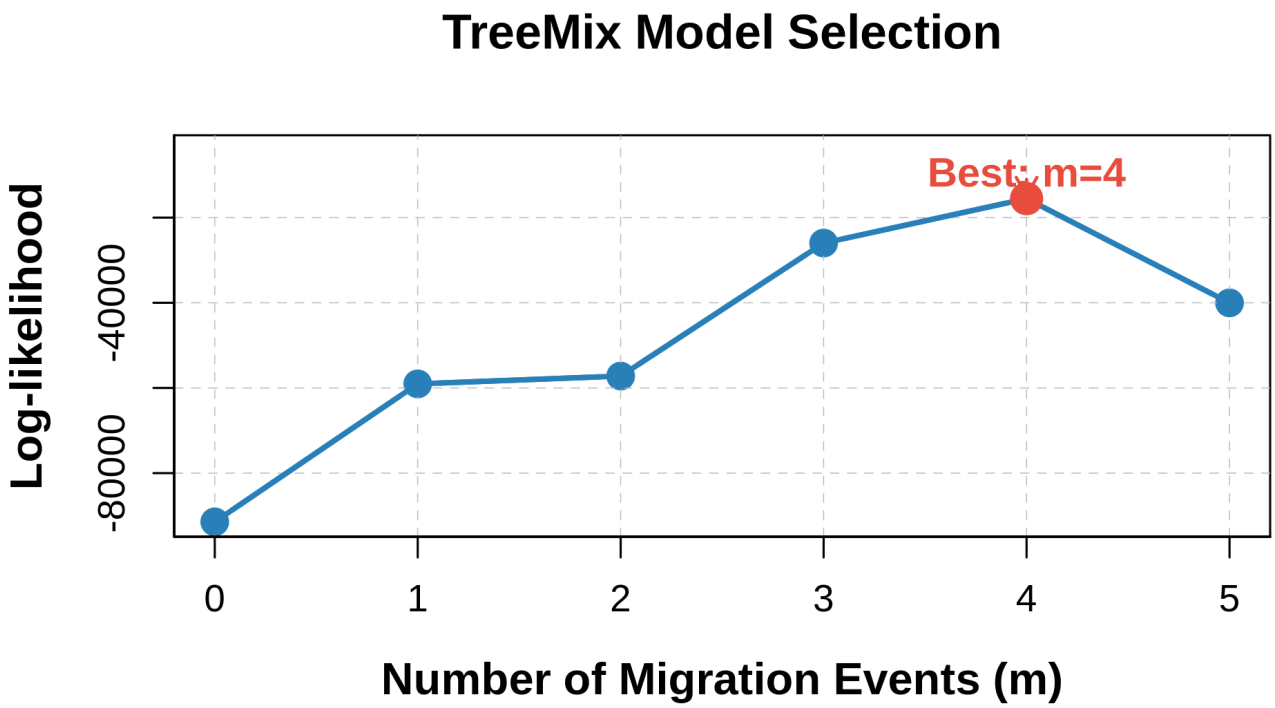
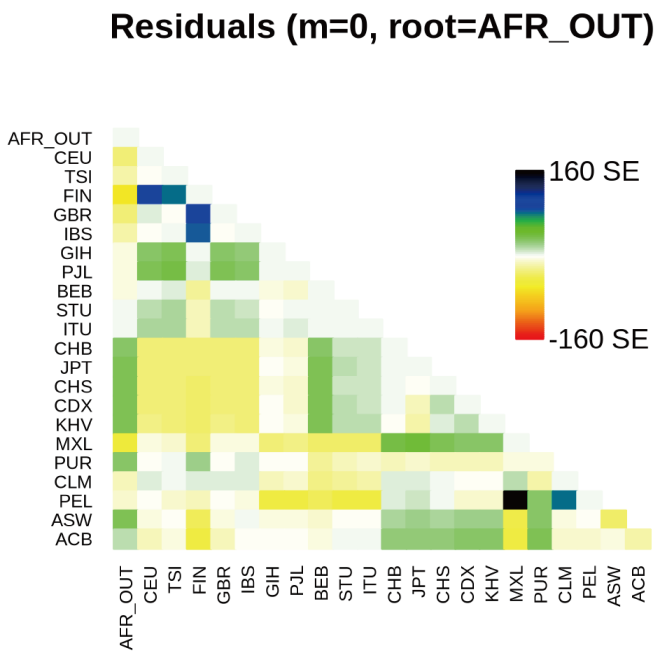


Figure 16 PopGenAgent-made. TreeMix model-selection curve with `root=AFR_OUT`. The log-likelihood is plotted against the number of migration edges (m); the red marker indicates the best-supported model ($m = 4$) under this scan. The curve is used to select the migration-edge complexity for subsequent TreeMix graph fitting and residual diagnostics. To enhance robustness, PopGenAgent performs repeated runs with different random seeds and searches for the optimal solution; in this experiment, 2,000 random seeds were explored.



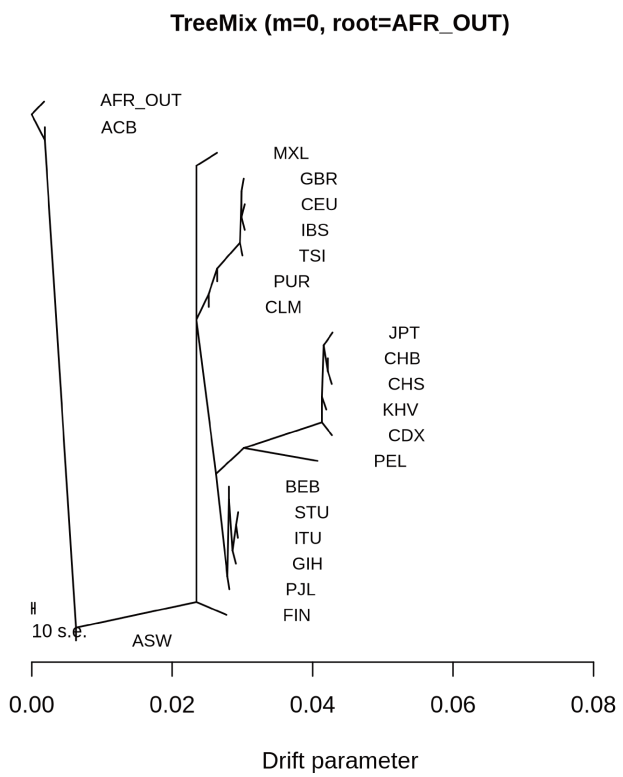
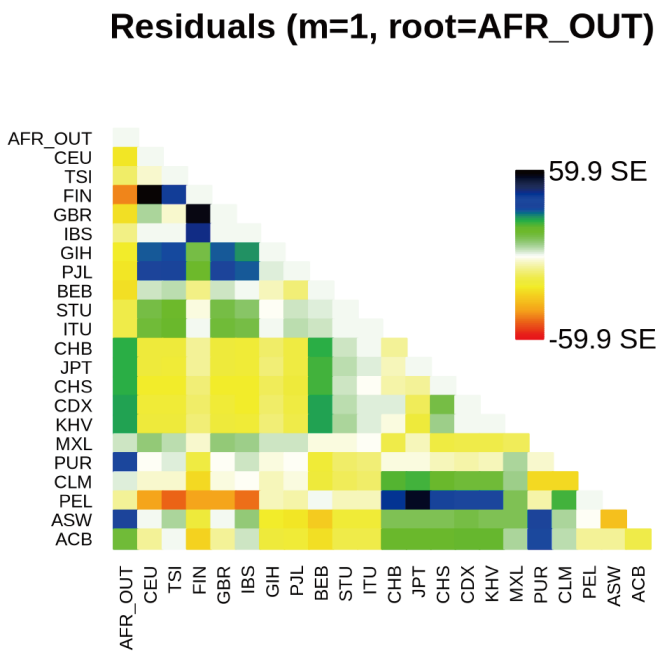


Figure 17 PopGenAgent-made. TreeMix model with zero migration edges ($m = 0$) using an aggregated outgroup setting (`AFR_OUT`), where YRI, LWK, GWD, MSL, and ESN are merged as an outgroup and excluded from the modeled set. The resulting maximum-likelihood tree summarizes major population splits without admixture edges. Residual covariance indicates remaining departures from the tree model and motivates follow-up runs with migration edges reported elsewhere.



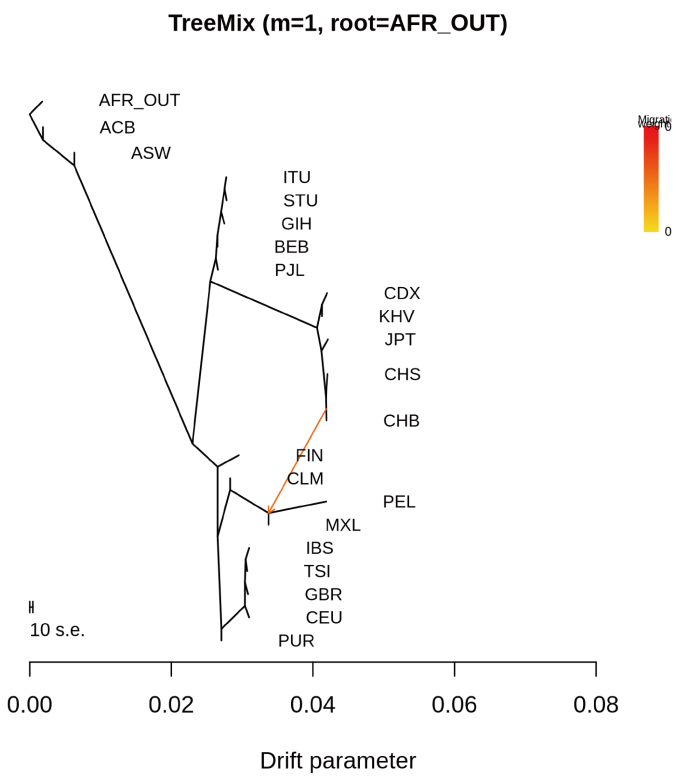
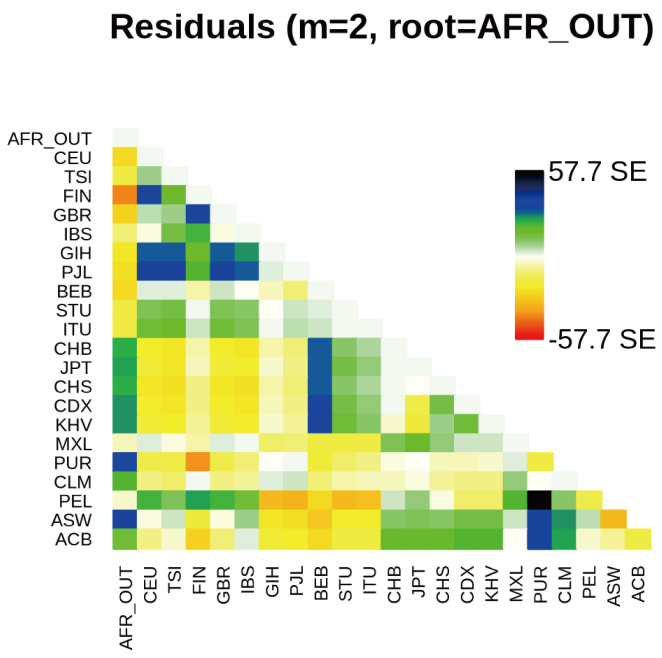


Figure 18 PopGenAgent-made. TreeMix model with one migration edges ($m = 1$) using an aggregated outgroup setting (`AFR_OUT`), where YRI, LWK, GWD, MSL, and ESN are merged as an outgroup and excluded from the modeled set.



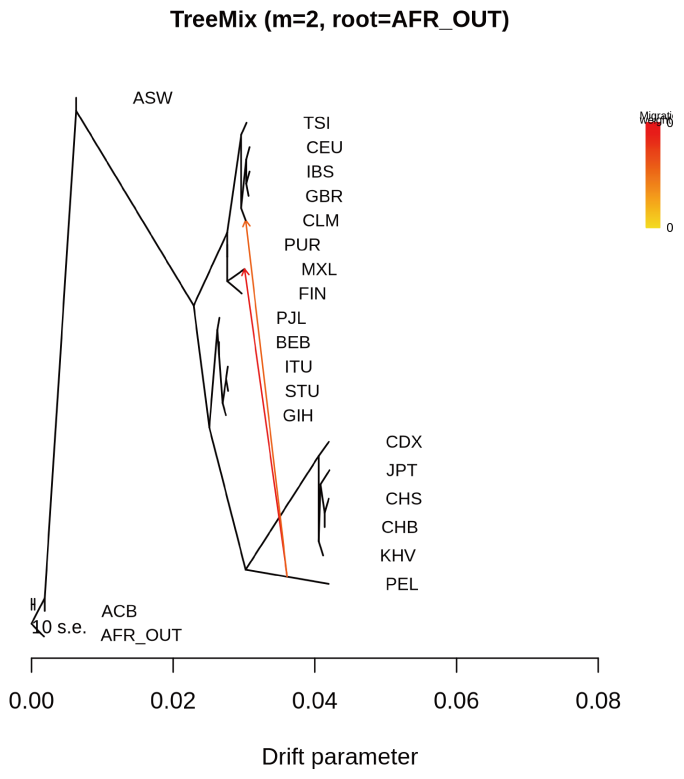
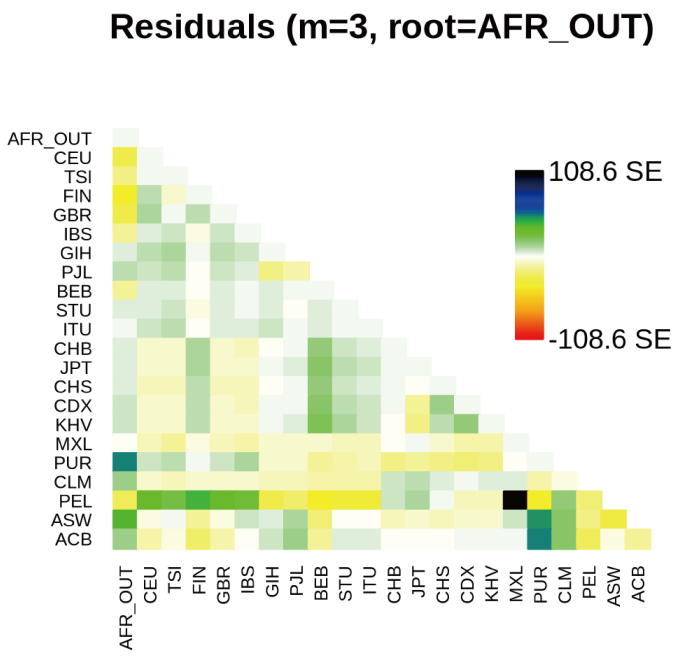


Figure 19 PopGenAgent-made. TreeMix model with two migration edges ($m = 2$) using an aggregated outgroup setting (`AFR_OUT`), where YRI, LWK, GWD, MSL, and ESN are merged as an outgroup and excluded from the modeled set.



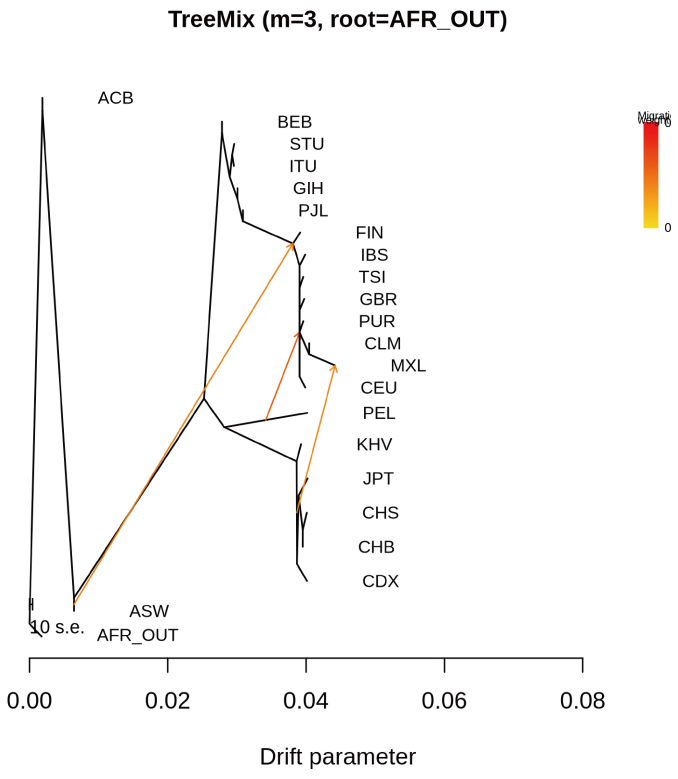
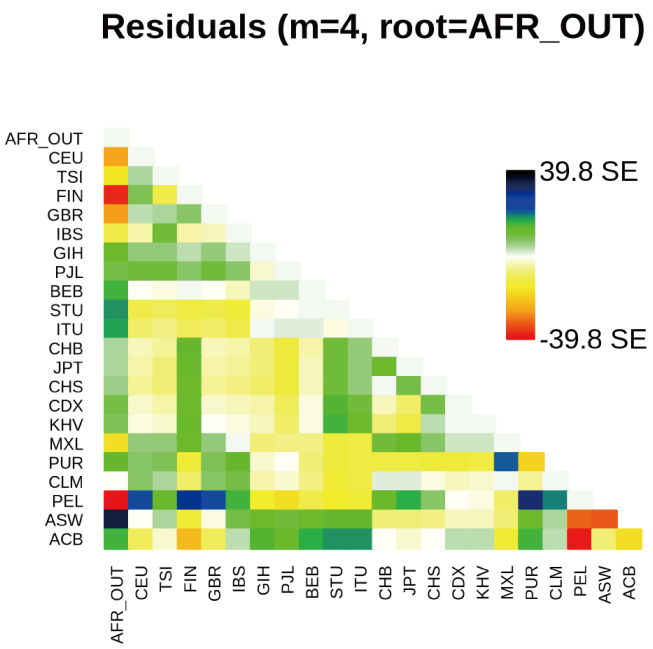


Figure 20 PopGenAgent-made. TreeMix model with three migration edges ($m = 3$) using an aggregated outgroup setting (`AFR_OUT`), where YRI, LWK, GWD, MSL, and ESN are merged as an outgroup and excluded from the modeled set.



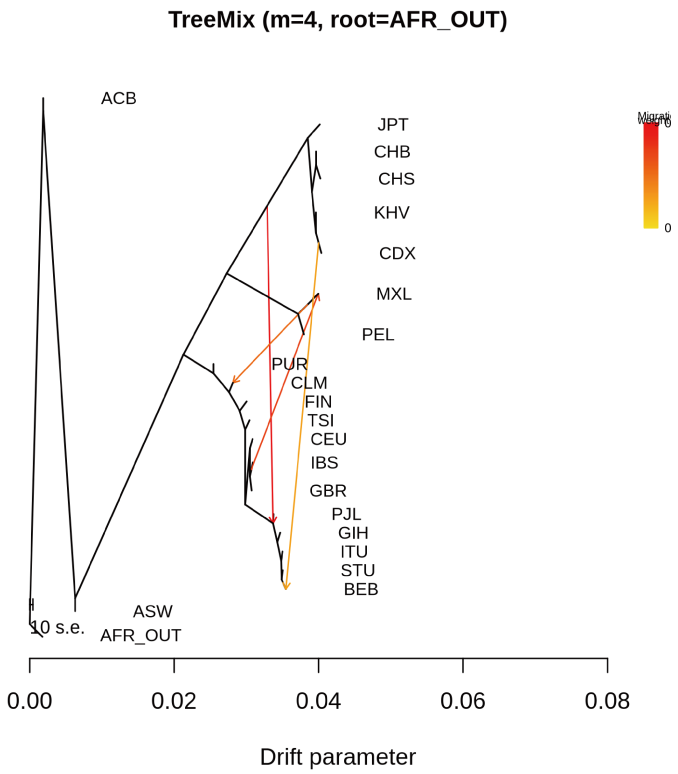
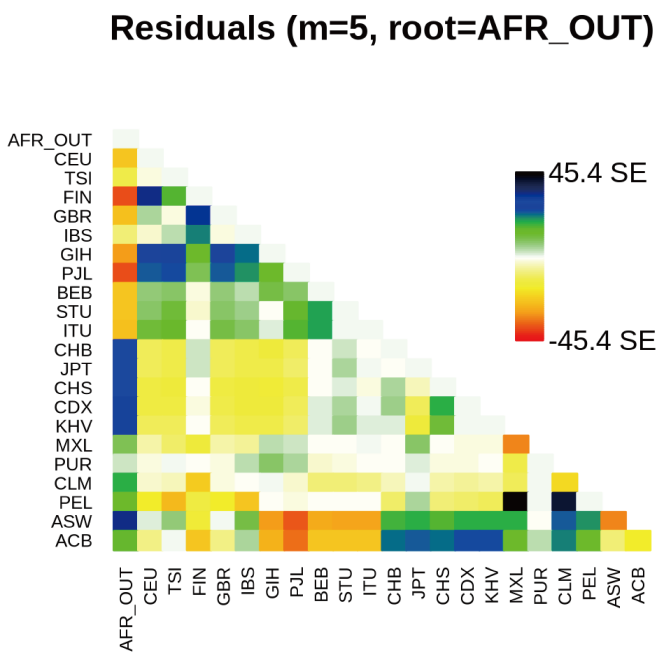


Figure 21 PopGenAgent-made. TreeMix model with four migration edges ($m = 4$) using an aggregated outgroup setting (`AFR_OUT`), where YRI, LWK, GWD, MSL, and ESN are merged as an outgroup and excluded from the modeled set.



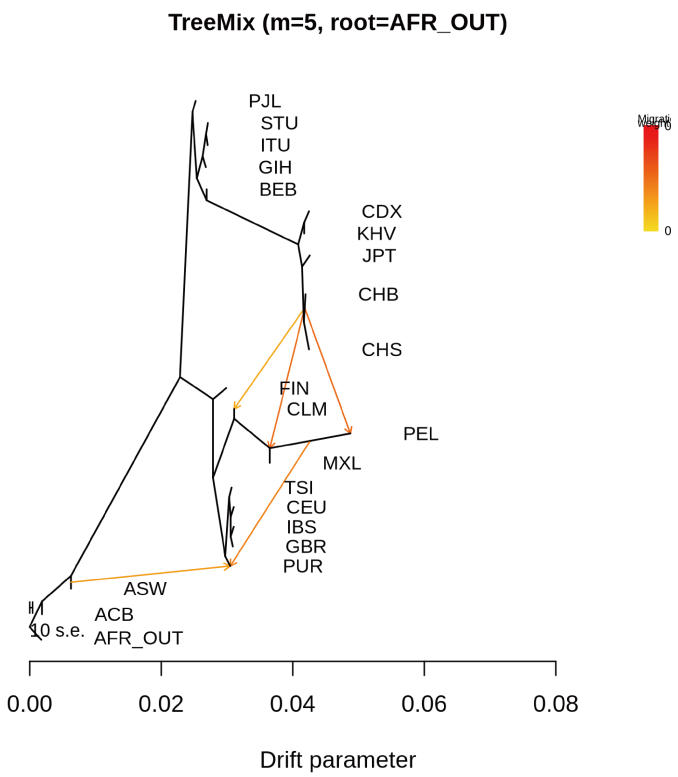


Figure 22 PopGenAgent-made. TreeMix model with five migration edges ($m = 5$) using an aggregated outgroup setting (`AFR_OUT`), where YRI, LWK, GWD, MSL, and ESN are merged as an outgroup and excluded from the modeled set.

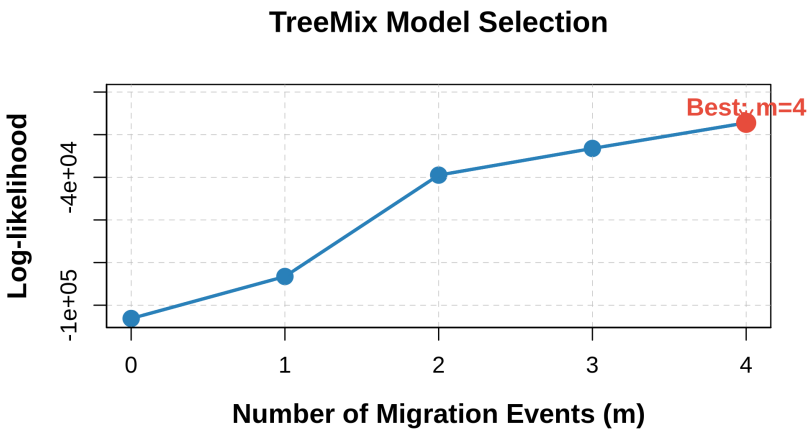


Figure 23 PopGenAgent-made. TreeMix model-selection curve for the same dataset under an alternative scan range ($m = 0$–4). Log-likelihood increases with added migration edges and attains its maximum at $m = 4$ (annotated), supporting $m = 4$ as the preferred complexity within the evaluated range.

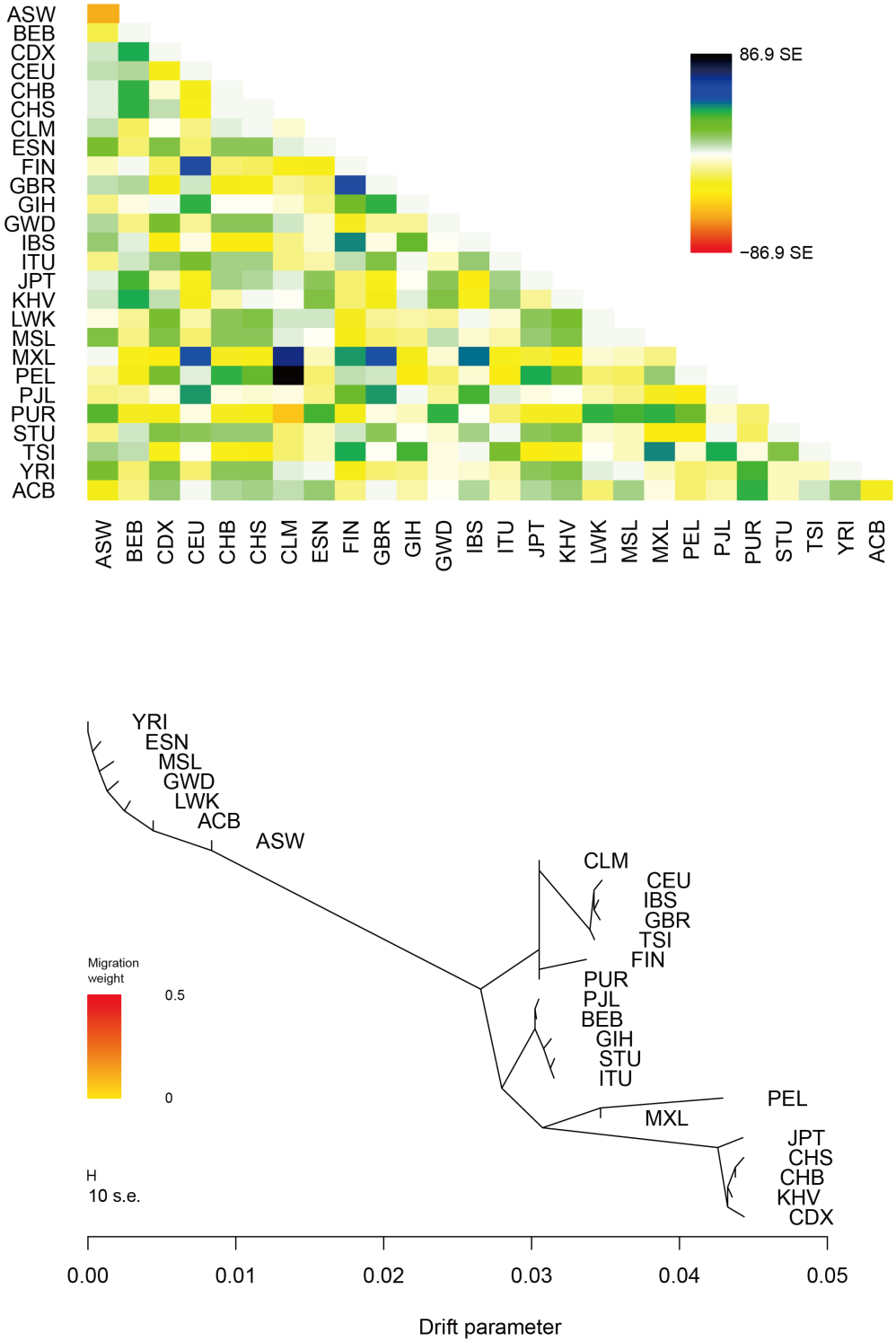


Figure 24 PopGenAgent-made. TreeMix model with zero migration edges ($m = 0$) under the original population set (no `AFR_OUT` exclusion), showing the maximum-likelihood population tree without admixture. The topology recapitulates major continental splits, with early divergence of African lineages. The residual covariance highlights population pairs that deviate from the fitted tree, consistent with allele-sharing patterns not captured by an $m = 0$ model.

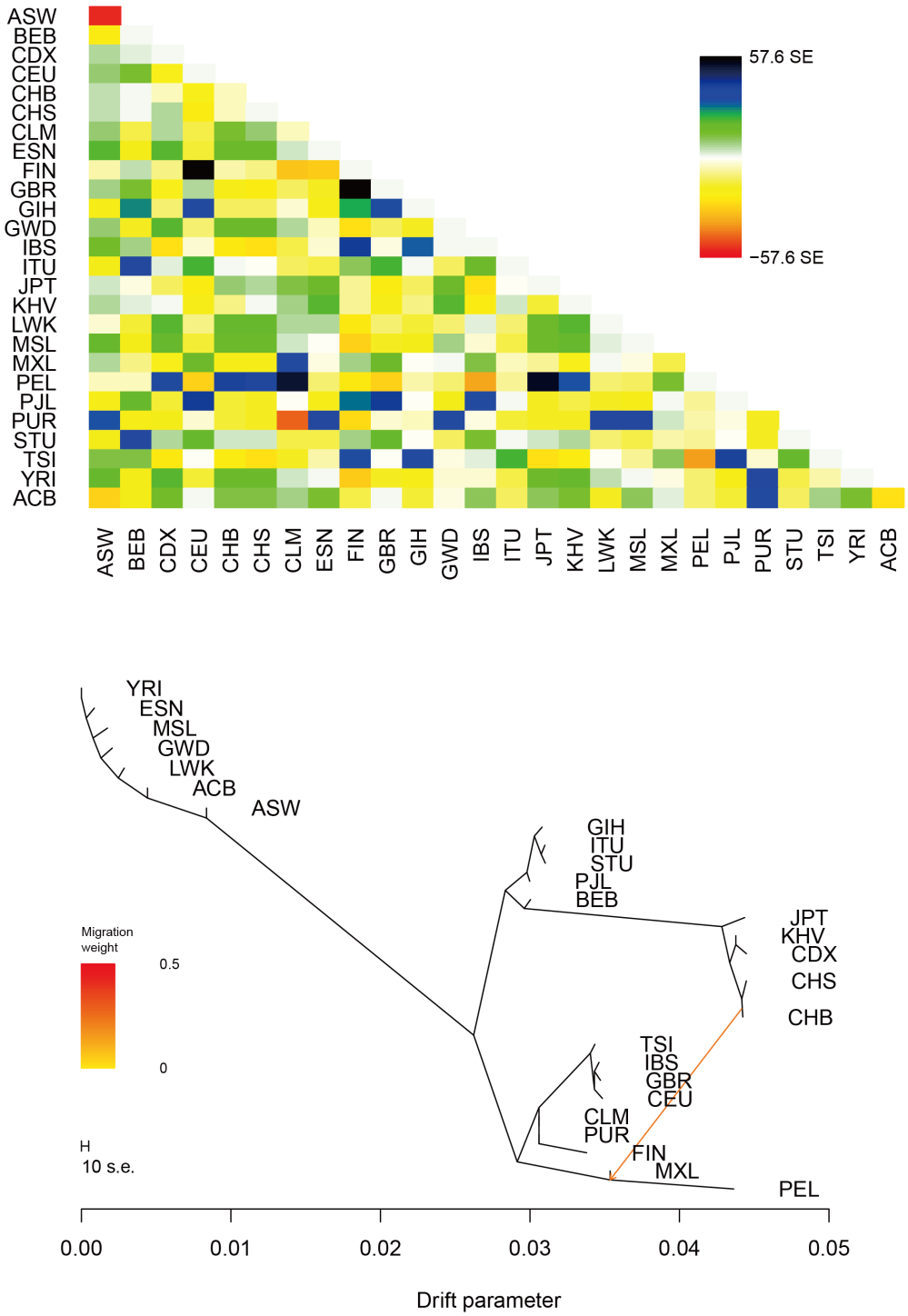


Figure 25 PopGenAgent-made. TreeMix model with one migration edge ($m = 1$) under the original population set (no AFR_OUT exclusion), capturing a single gene flow event from CHB populations into FIN. The added migration improves model likelihood and reduces residual error, particularly among admixed American populations.

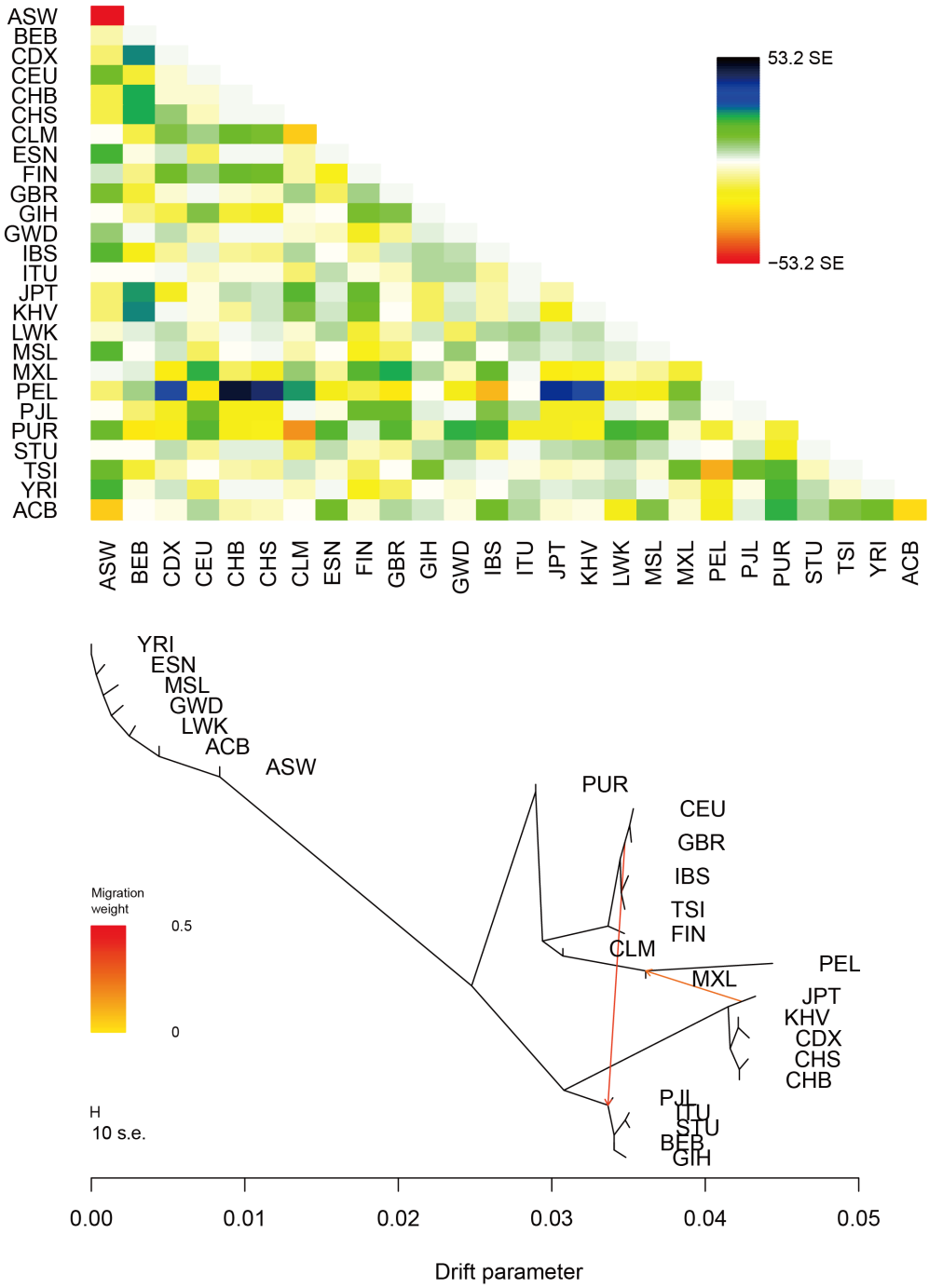


Figure 26 PopGenAgent-made. TreeMix model with two migration edges ($m = 2$) under the original population set (no AFR_OUT exclusion), adding an additional admixture signal between South Asian and European lineages. The model better explains the genetic structure of admixed groups and further reduces residuals.

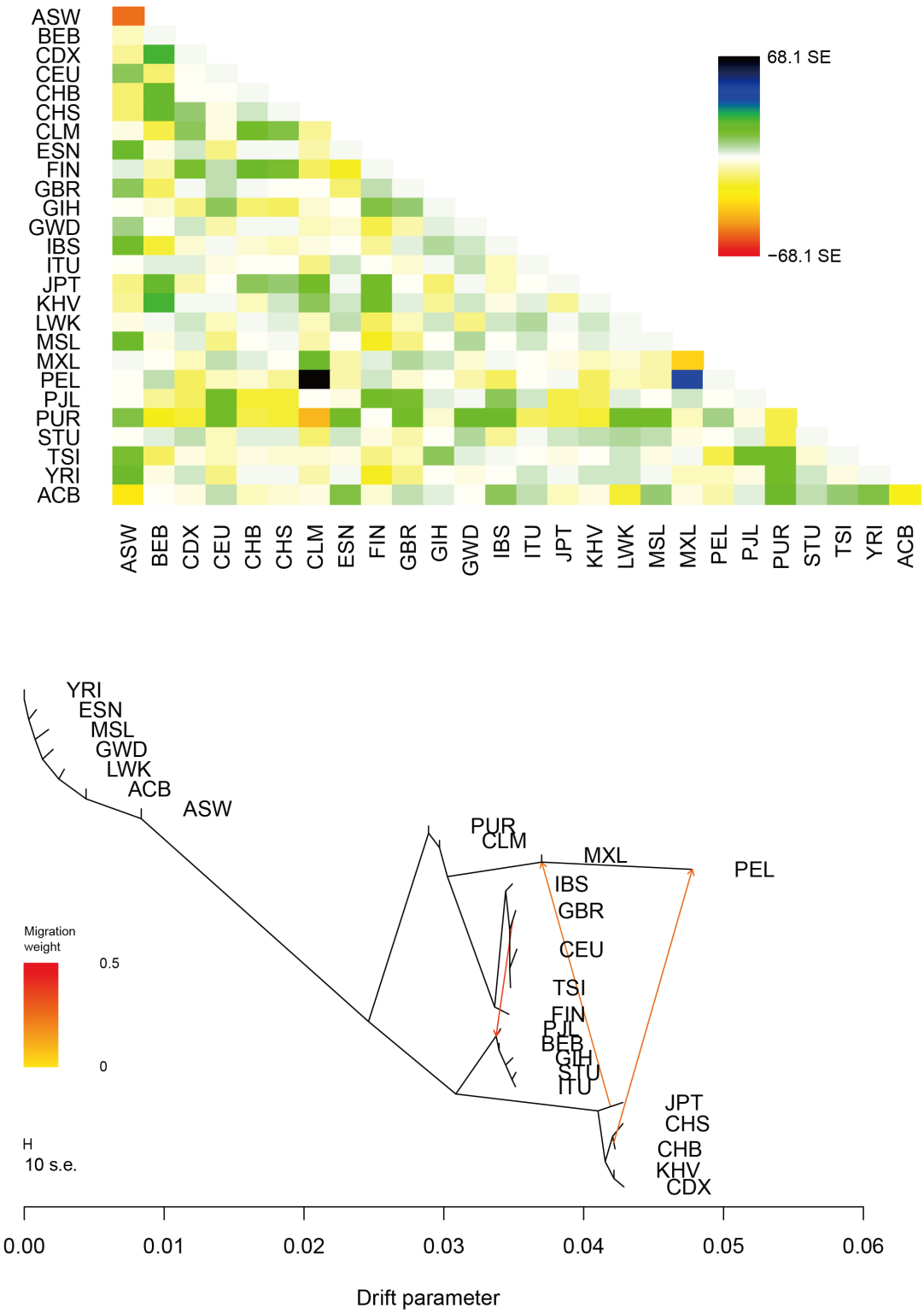


Figure 27 PopGenAgent-made. TreeMix model with three migration edges ($m = 3$) under the original population set (no AFR_OUT exclusion), introducing gene flow from European into South Asian, and East Asian into European populations. The inferred events reflect post-colonial admixture histories and enhance model fit.

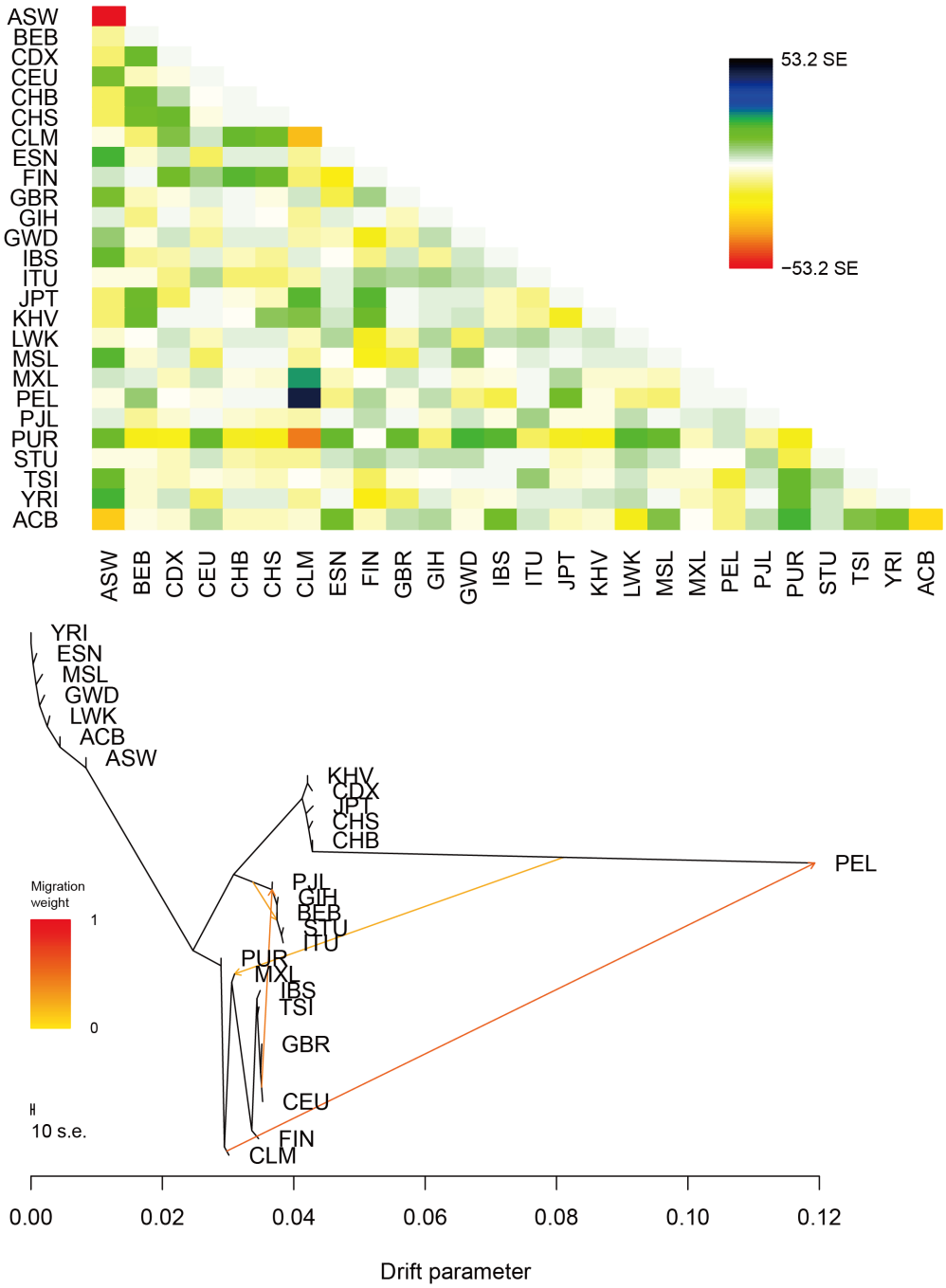


Figure 28 PopGenAgent-made. TreeMix model with four migration edges ($m = 4$) under the original population set (no `AFR_OUT` exclusion), capturing multiple intercontinental gene flow events, especially into PEL. This expanded model explains the majority of residual patterns, aligning with known demographic and historical admixture events.

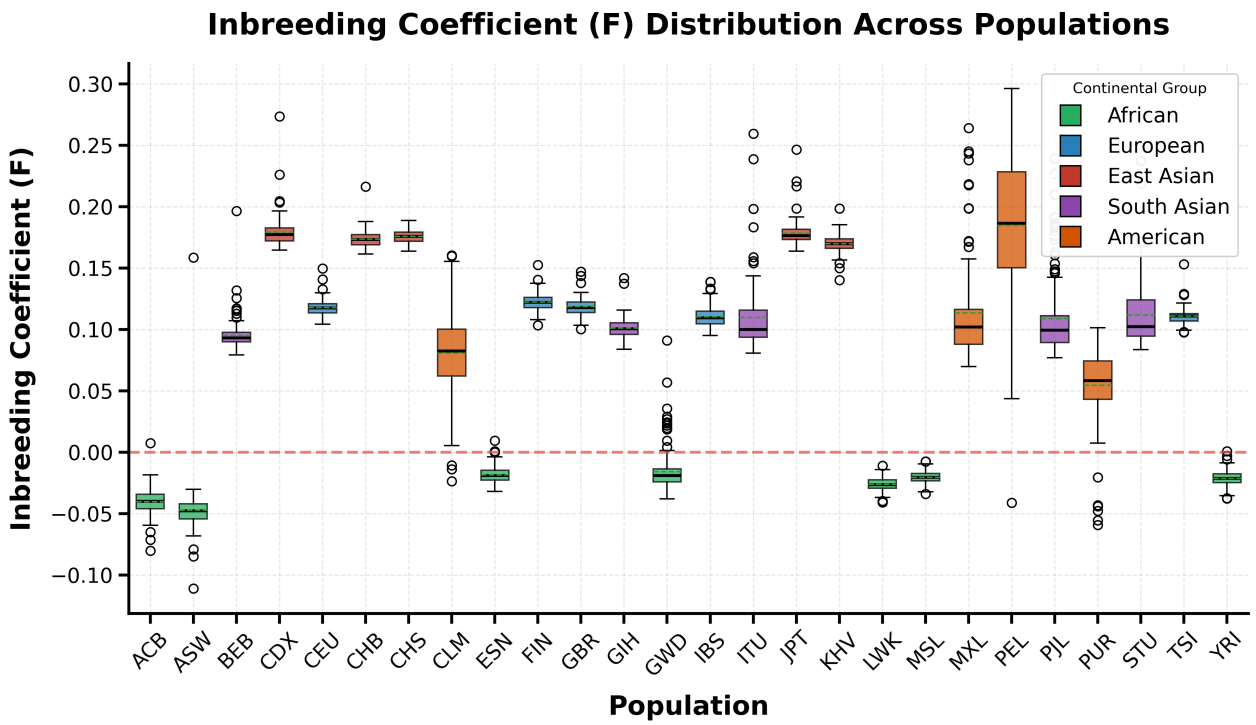


Figure 29 PopGenAgent-made. Distribution of individual inbreeding coefficients (F) across 26 populations, independently generated by PopGenAgent. Boxplots summarize within-population variation, with populations grouped and color-coded by continental ancestry.

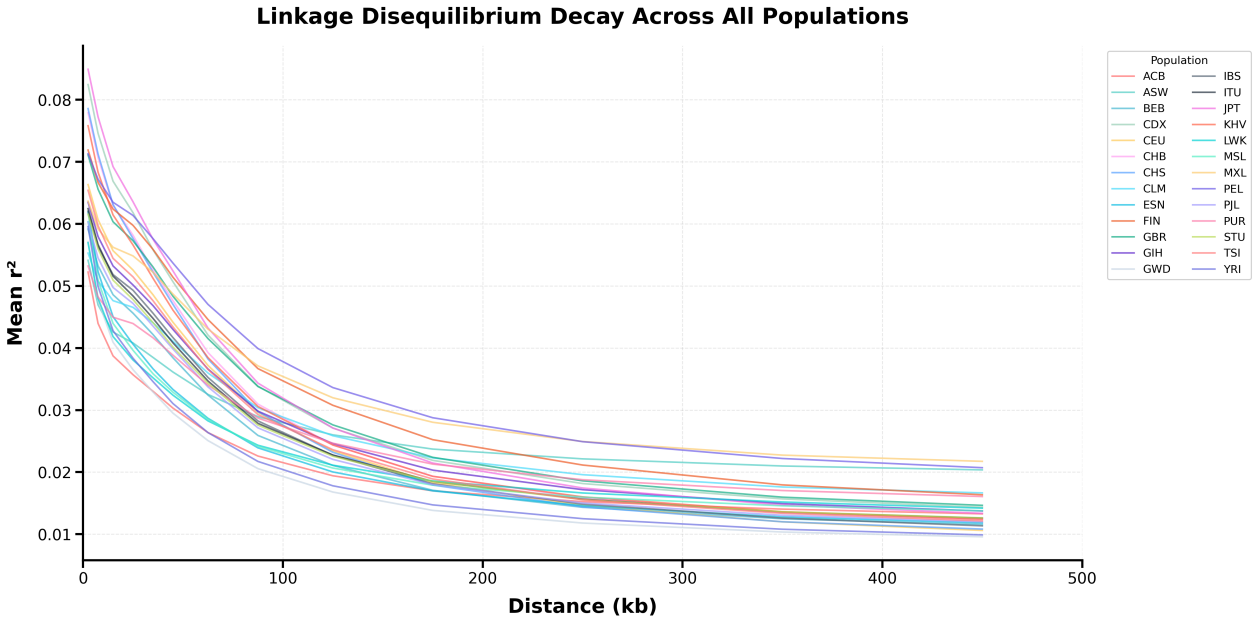


Figure 30 PopGenAgent-made. Linkage disequilibrium (LD) decay curves for all 26 populations, generated in an independent repeat by PopGenAgent. Mean pairwise r^2 is plotted as a function of physical distance (kb), with each curve representing one population.

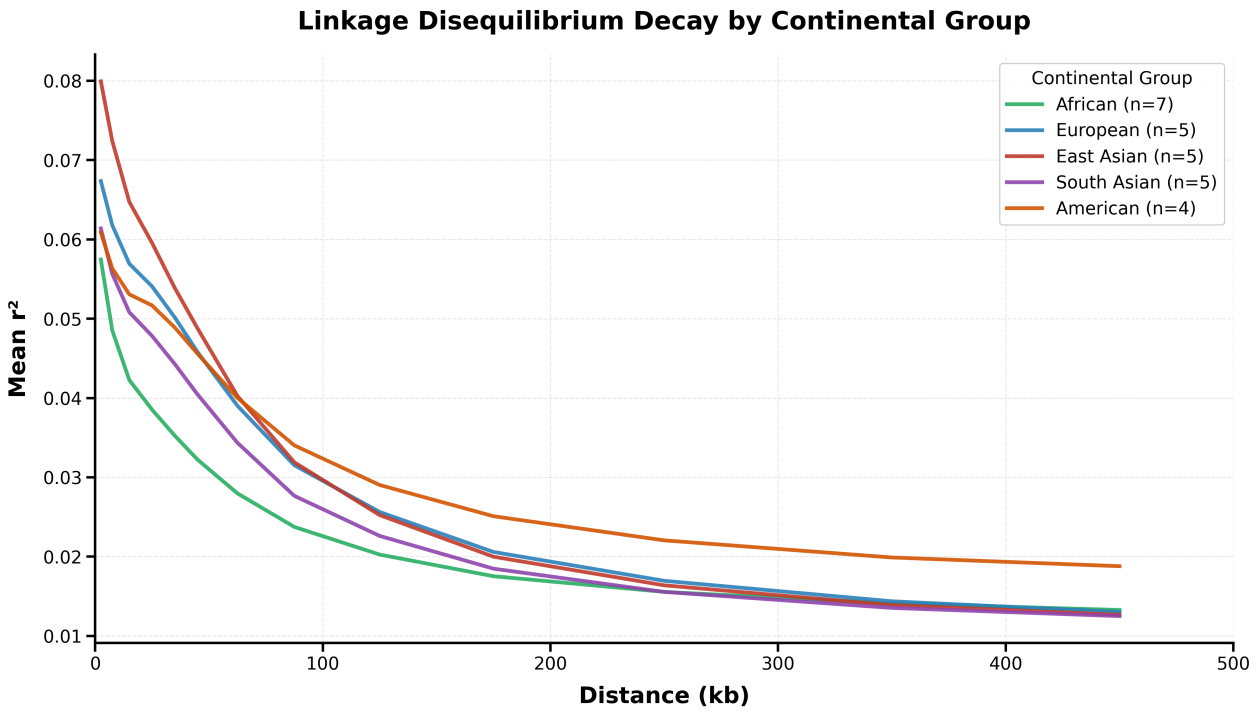


Figure 31 PopGenAgent-made. Average linkage disequilibrium decay stratified by continental group, produced by PopGenAgent in an independent run. Curves show mean r^2 as a function of distance, averaged across populations within each continental group.

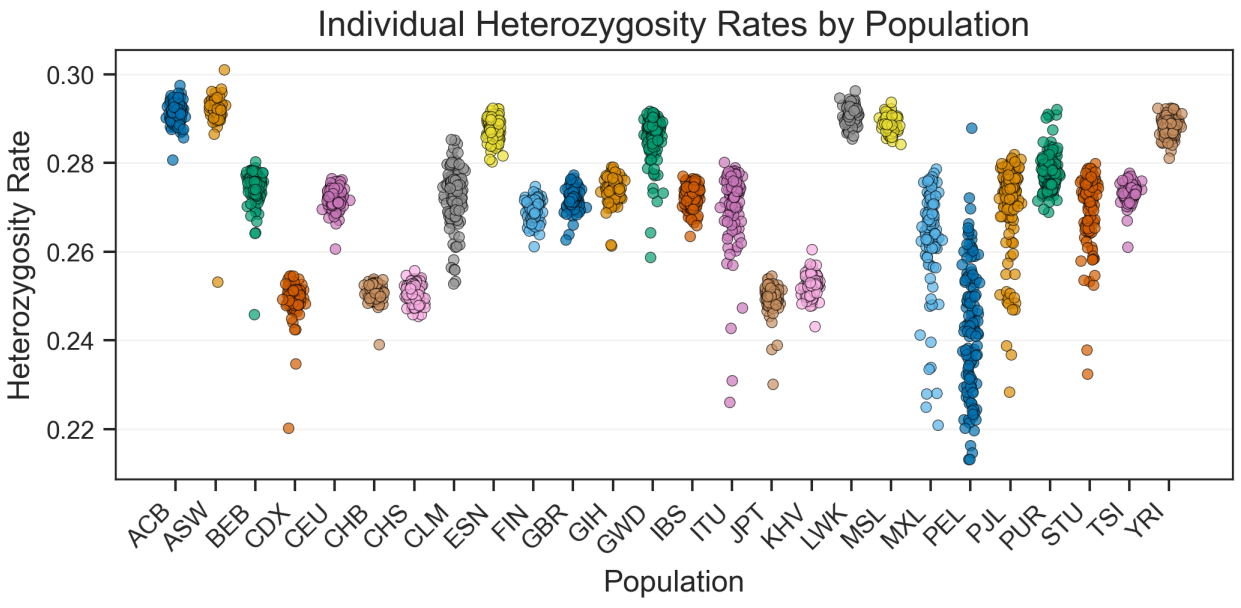


Figure 32 PopGenAgent-made. Heterozygosity patterns across 1000 Genomes populations, automatically generated in a repeated experiment without manual curation. Includes a per-individual heterozygosity scatter plot to visualize within-population diversity.

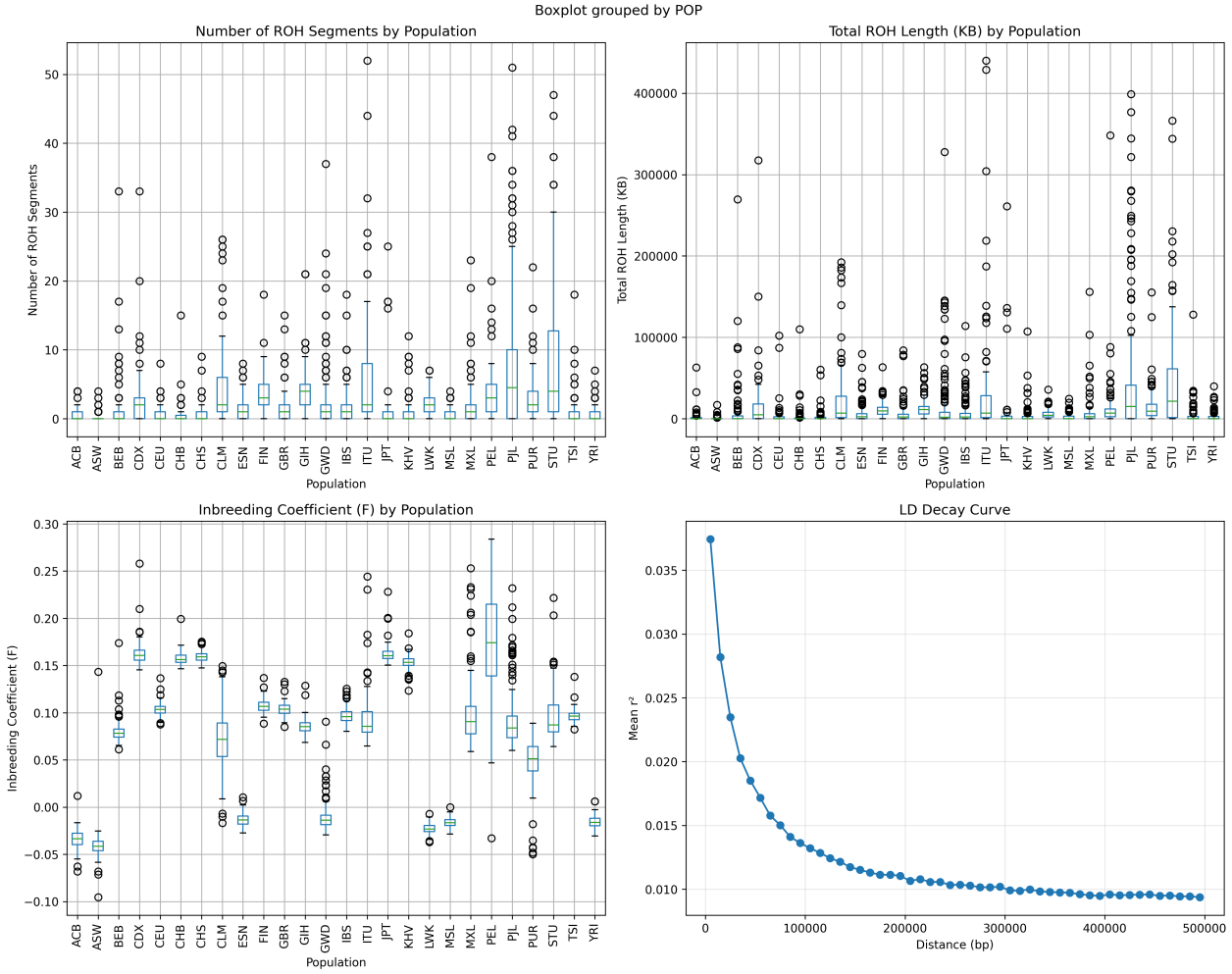


Figure 33 PopGenAgent-made. Summary of population-level metrics automatically generated by PopGenAgent without human intervention. The figure includes boxplots of the number and total length of runs of homozygosity (ROH), inbreeding coefficients (F), and an LD decay curve. The LD decay was erroneously computed across all 26 populations jointly, reflecting a limitation of the fully automated pipeline in context-specific stratification.

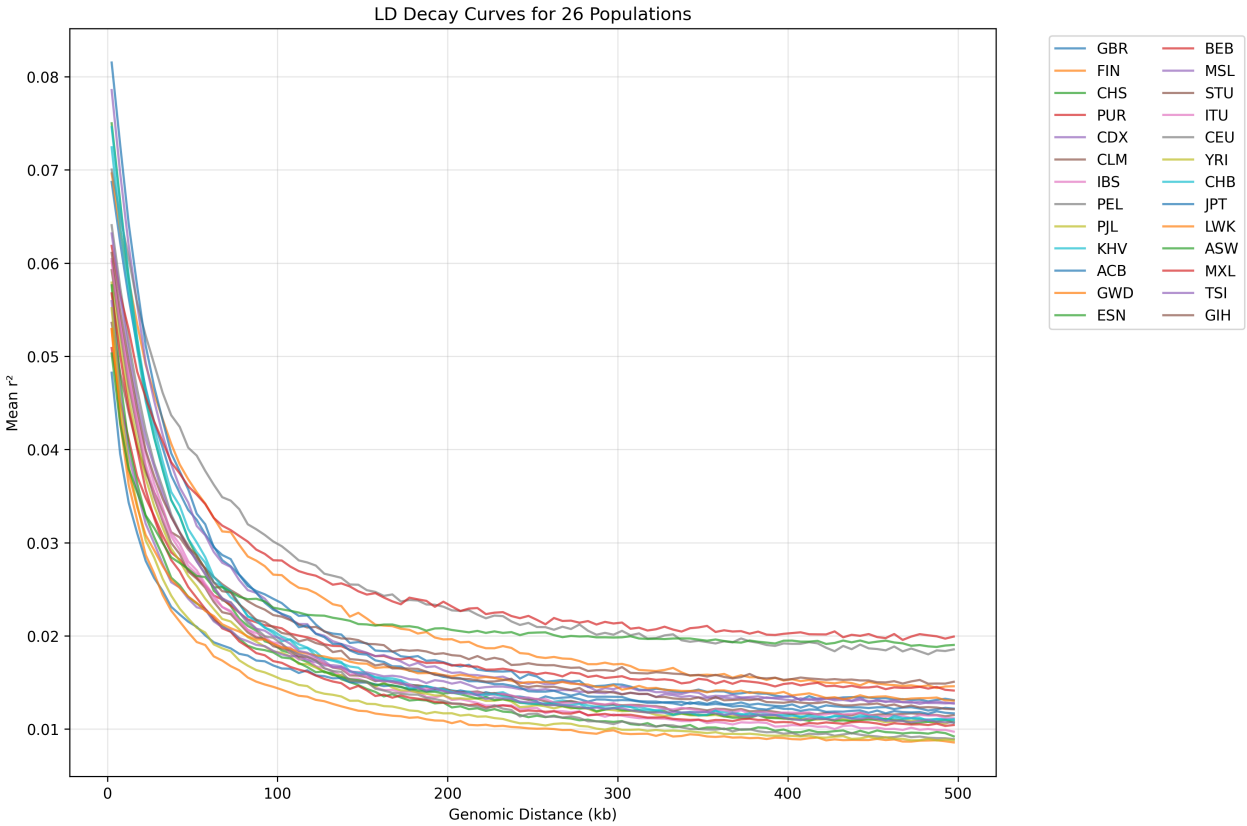


Figure 34 PopGenAgent-made. LD decay curves for all 26 populations in the 1000 Genomes Project. Despite overly expanded Y-axis scaling due to a lack of axis estimation in the automated workflow, the underlying r^2 decay patterns accurately reflect known population differences.

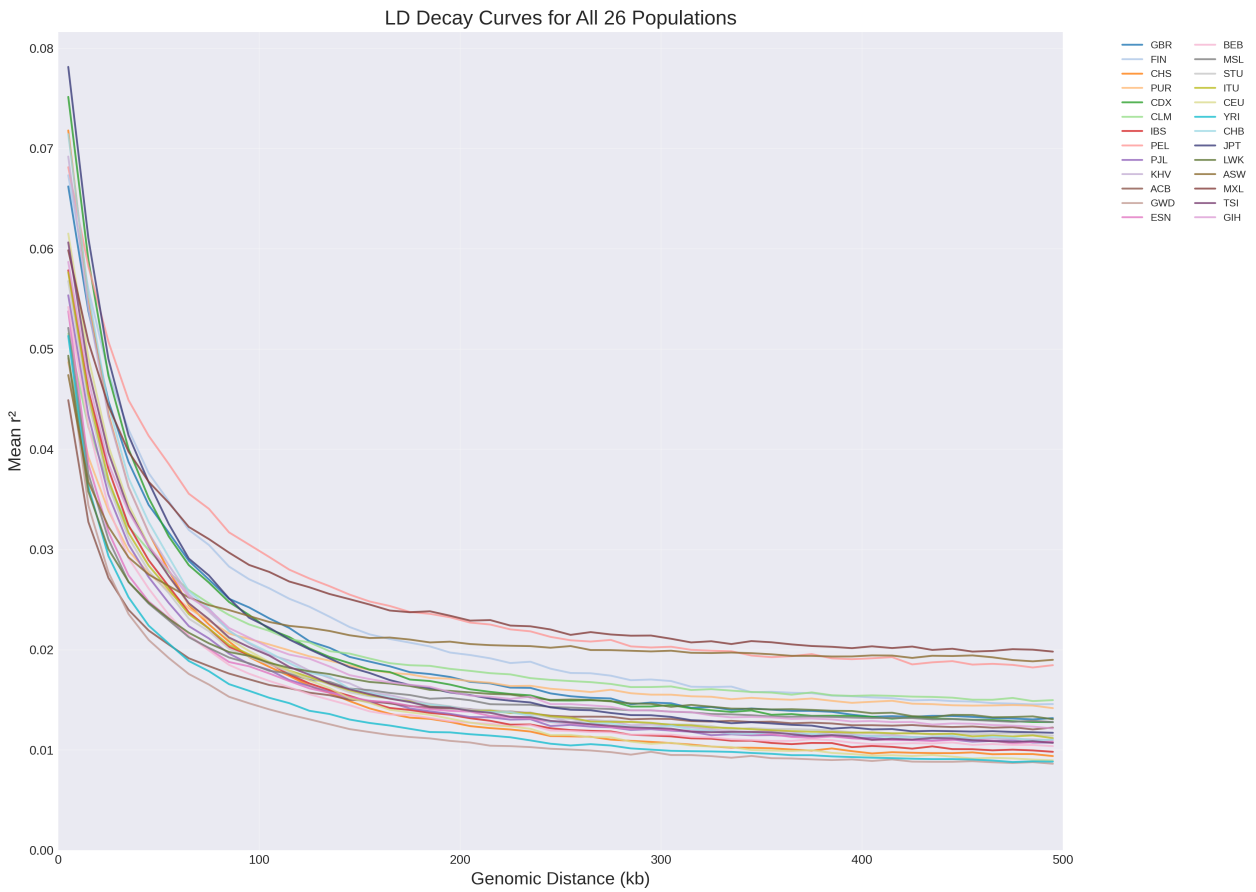


Figure 35 PopGenAgent-made. LD decay curves for 26 populations in the 1000 Genomes Project, generated in a repeated experiment by PopGenAgent. The output preserves biologically meaningful patterns, and the Y-axis was appropriately scaled without manual adjustment.

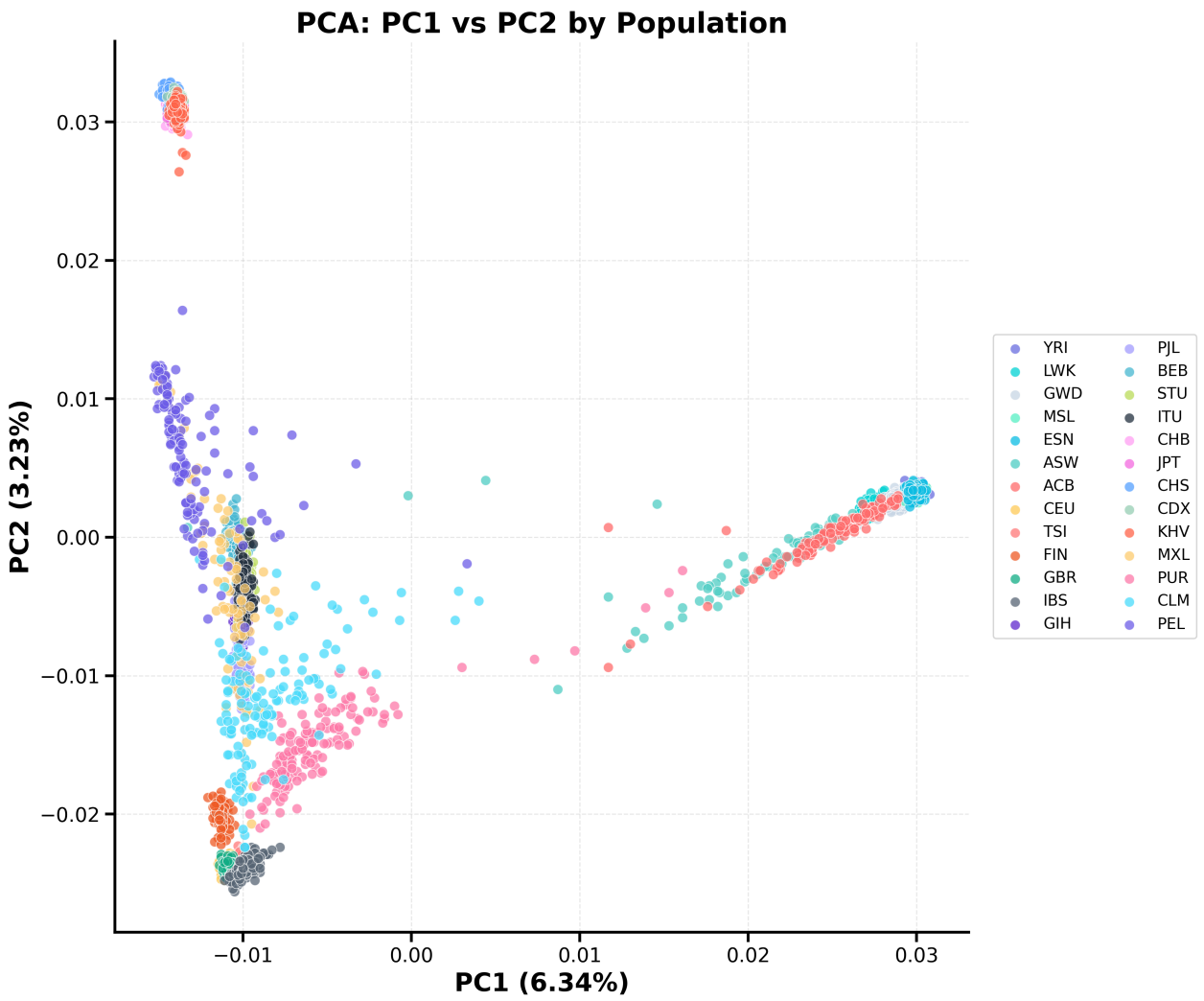


Figure 36 PopGenAgent-made. Principal component analysis (PCA) of 26 populations generated by PopGenAgent. The panel shows PC1 versus PC2 projections, with individuals colored by population labels; percentages indicate variance explained by each component.

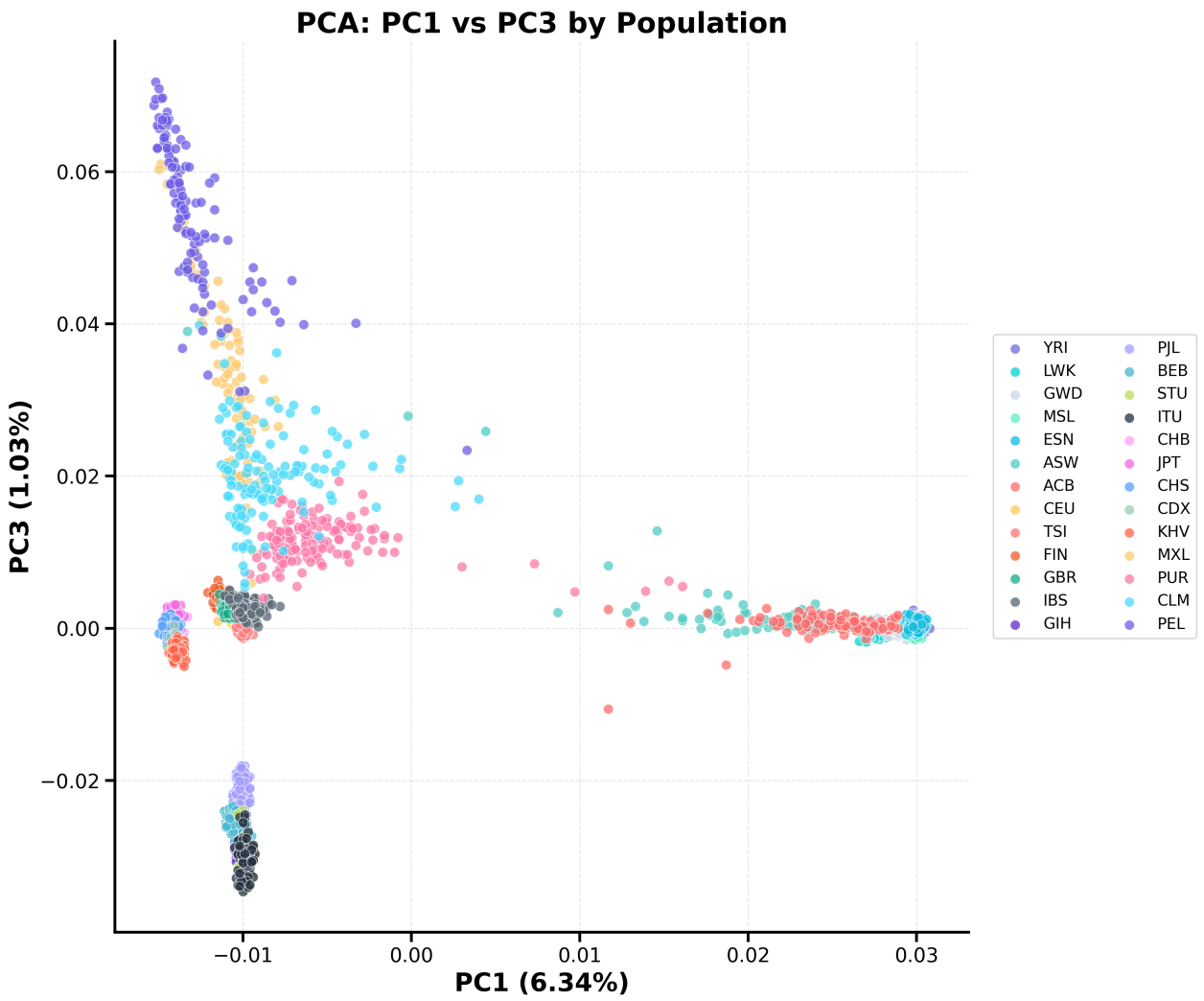


Figure 37 PopGenAgent-made. Principal component analysis (PCA) of 26 populations generated by PopGenAgent. The panel shows PC1 versus PC3 projections, with individuals colored by population labels; percentages indicate variance explained by each component.

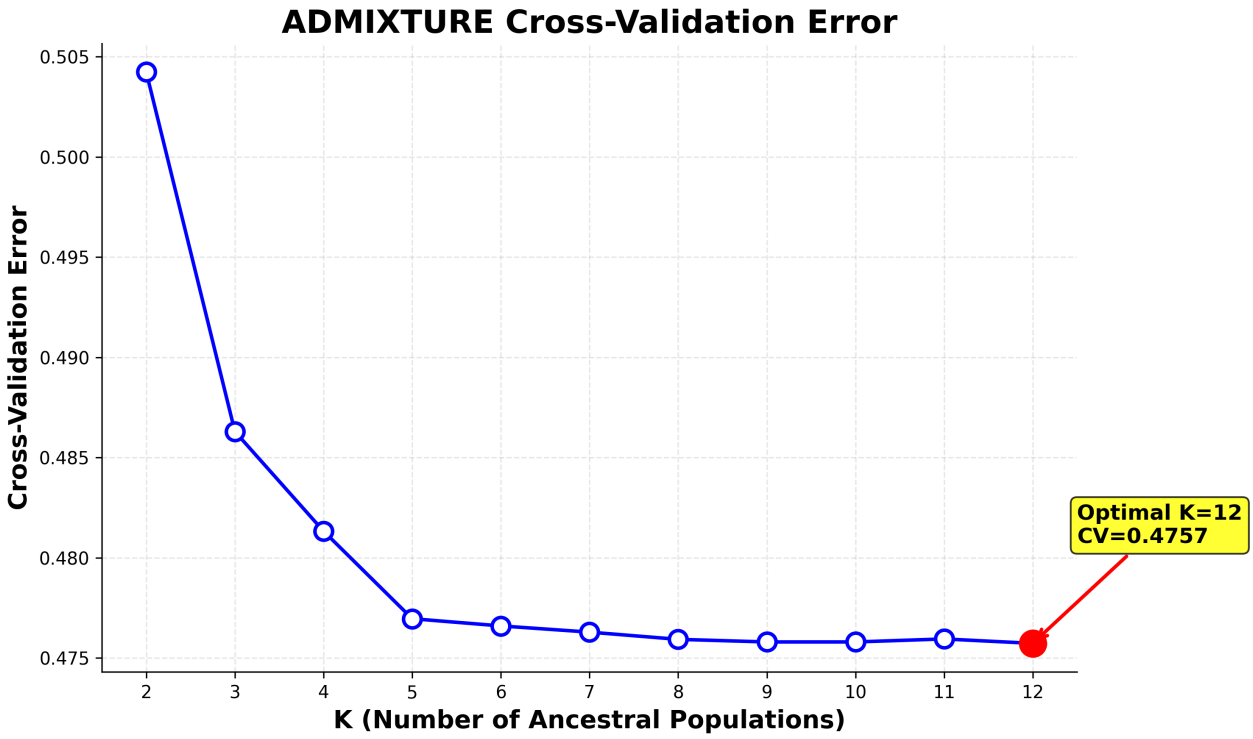


Figure 38 PopGenAgent-made. ADMIXTURE ancestry inference for 26 populations across multiple values of K, generated by PopGenAgent. Each vertical bar represents an individual, grouped by population and colored by inferred ancestry components.

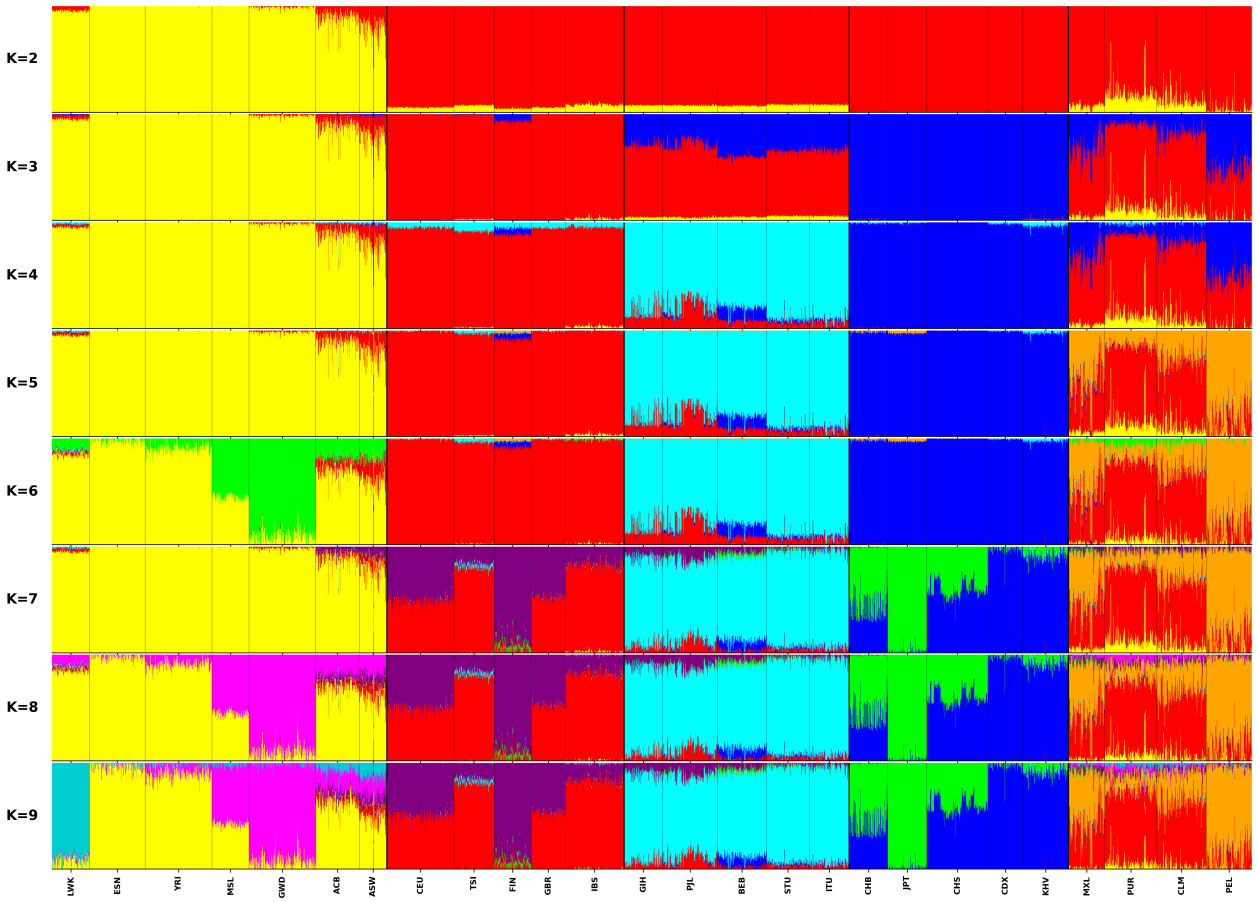


Figure 39 PopGenAgent-made. ADMIXTURE cross-validation error as a function of ancestral population number K, independently computed by PopGenAgent. The minimum cross-validation error identifies the optimal K.

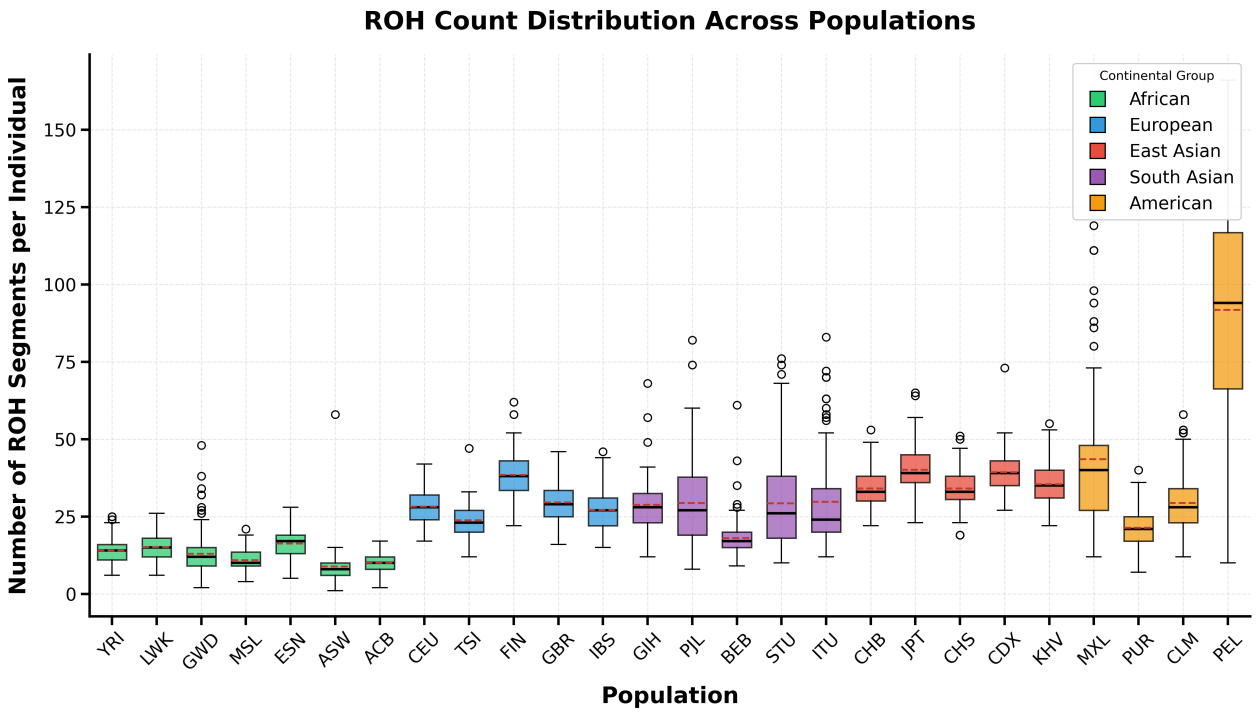


Figure 40 PopGenAgent-made. Distribution of runs of homozygosity (ROH) counts per individual across 26 populations, independently generated by PopGenAgent. Boxplots summarize individual-level ROH counts, with populations grouped and color-coded by continental ancestry.

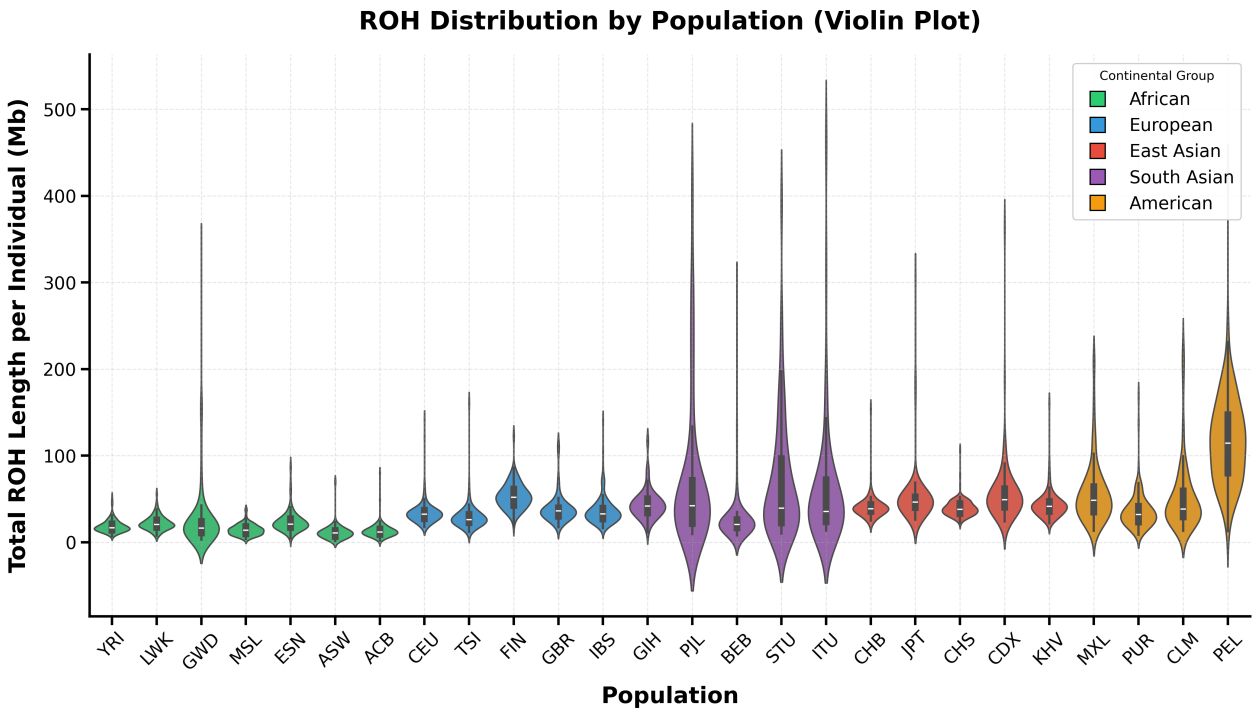


Figure 41 PopGenAgent-made. Violin plot showing the distribution of total ROH length per individual across 26 populations, independently produced by PopGenAgent. The density shape reflects population-specific variation in ROH burden, with colors indicating continental groups.

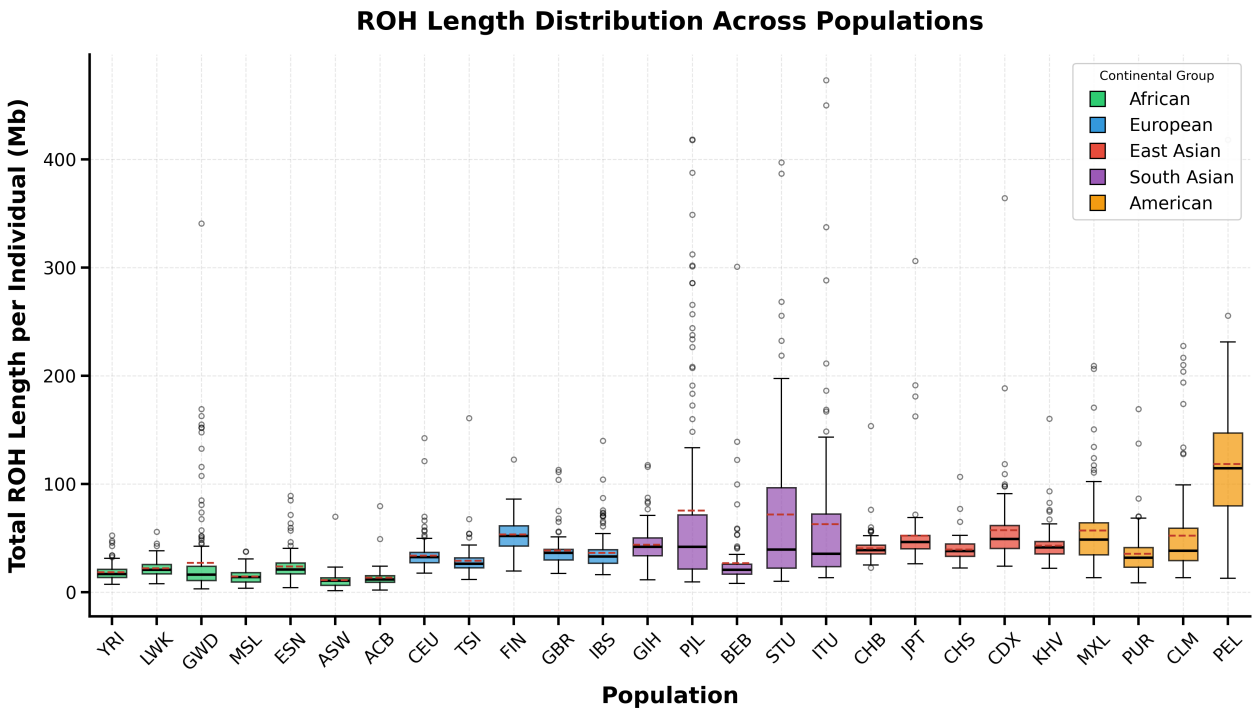


Figure 42 PopGenAgent-made. Distribution of total ROH length per individual across 26 populations, generated in an independent repeat by PopGenAgent. Boxplots depict variation in cumulative ROH length (Mb) within each population, grouped by continental ancestry.

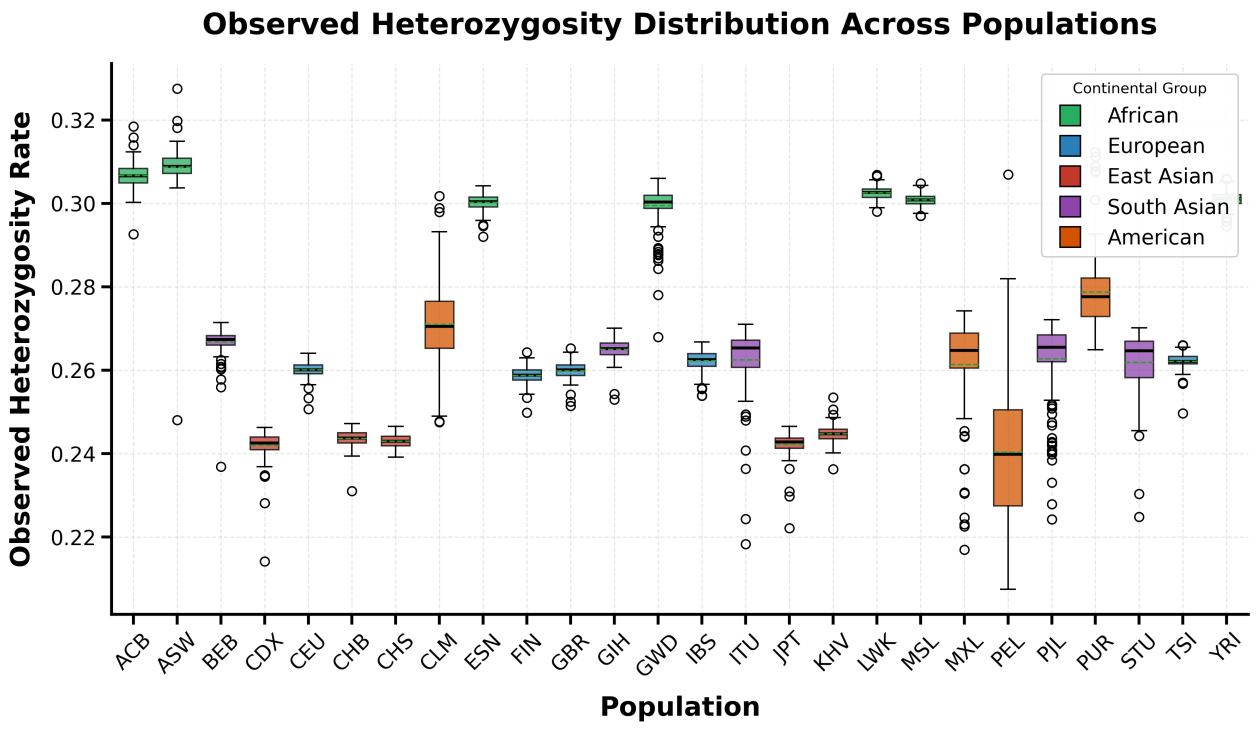


Figure 43 PopGenAgent-made. Distribution of observed heterozygosity across 26 populations, independently generated by PopGenAgent. Boxplots summarize individual-level variation, with populations grouped by continental ancestry.

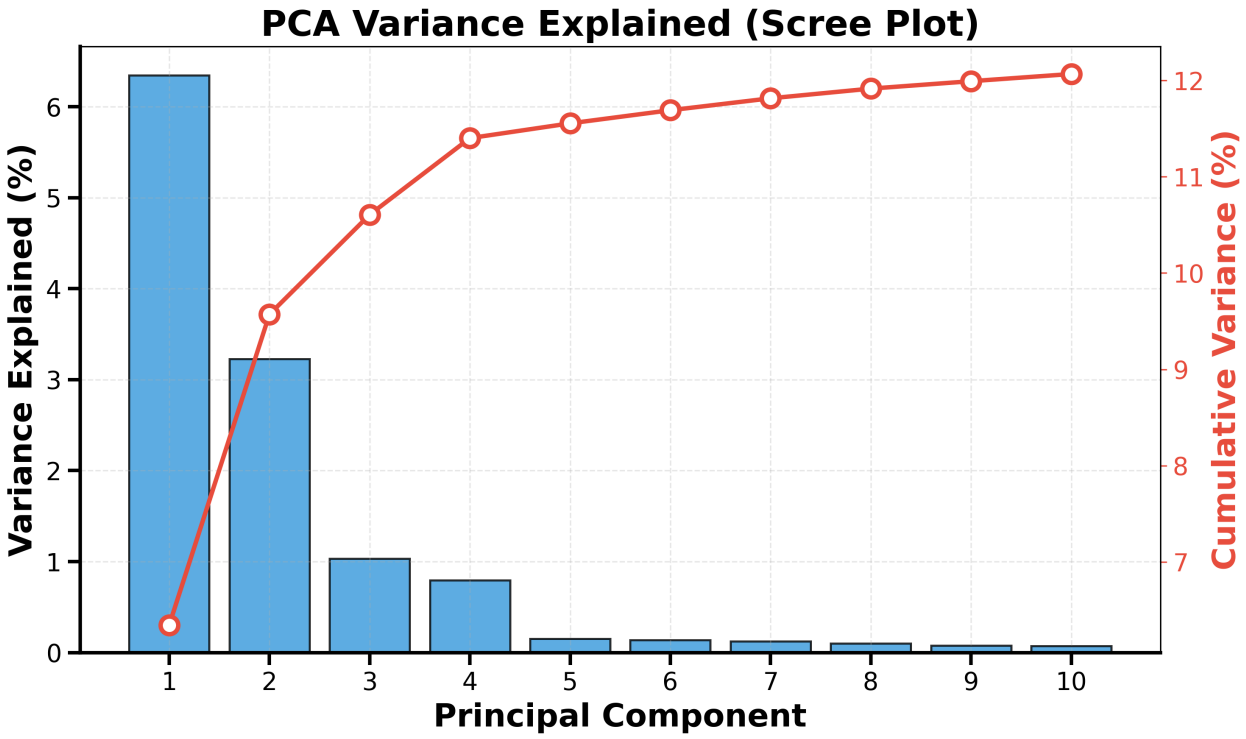


Figure 44 PopGenAgent-made. Scree plot of PCA variance explained, generated by PopGenAgent. Bars indicate the proportion of variance explained by each principal component, with the cumulative variance shown as a line.

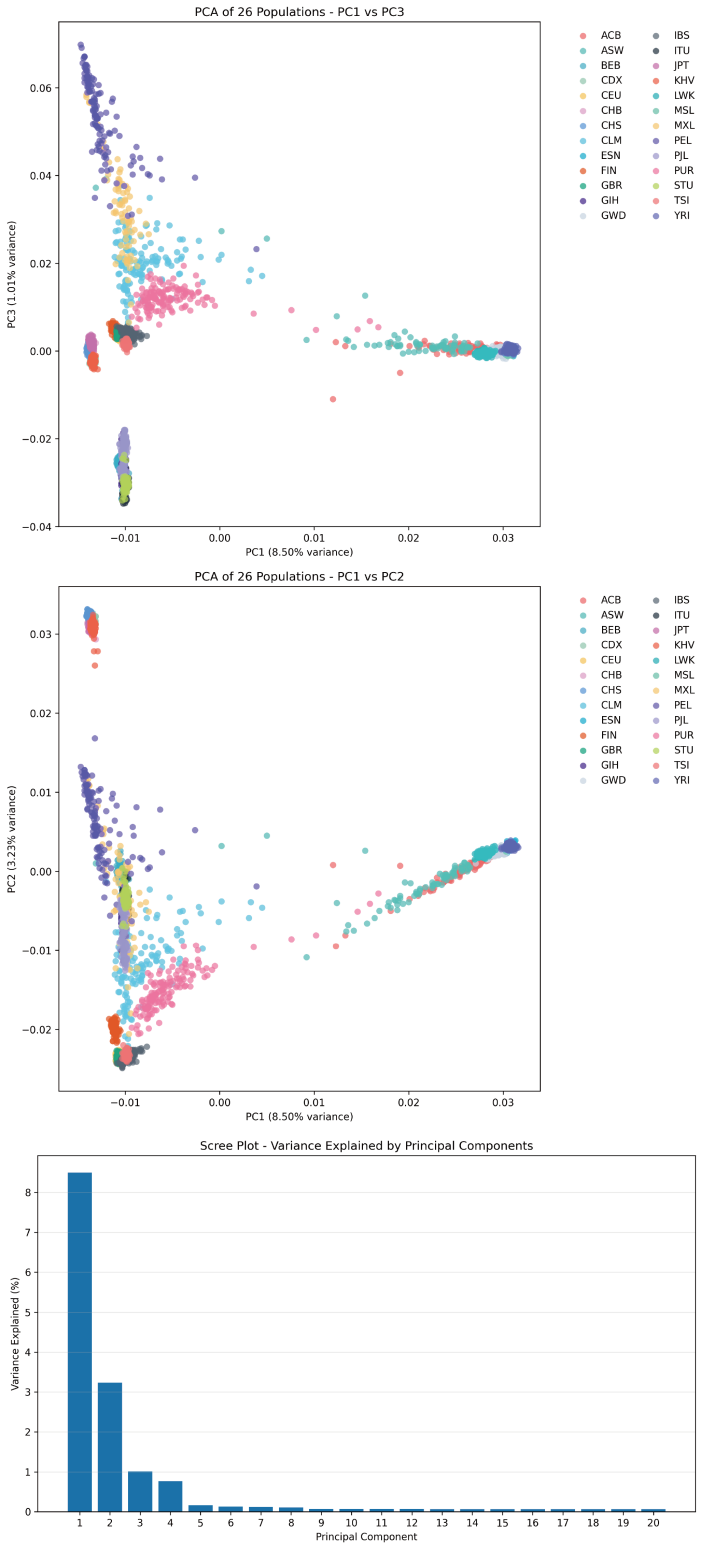


Figure 45 PopGenAgent-made. smartPCA results for 26 populations, generated in an independent repeat by PopGenAgent. The top panels show PC1 vs PC3 and PC1 vs PC2 projections, colored by population labels.