A057 · Composable foundations for agentic genomics

Presenter: Nezar Abdennur — UMass Chan Medical School

Authors: Nezar Abdennur, Conrad Bzura, Marek Wiewiorka, Vedat Yilmaz, Tomasz Gambin, Jacob Schreiber

Session: Day 1 · Thu Oct 1, 2026 · 2:15–4:15 PM

Autonomous agents are rapidly becoming credible collaborators in computational biology, but their competence is bounded by the tools and interfaces given to them. For genomics, that tooling is fragmented, brittle, and tightly coupled to specific file formats and line-oriented processing. Consequently, most bioinformatics code is dedicated to I/O, and most time is wasted routing data through tools that create intermediate copies at every step, and handling edge cases/deviations with ad hoc scripts. These influences not only bottleneck traditional analyses but also increasingly limit the scalability of training genomic machine learning (ML) models and usability by agents, who, like humans, must battle with the idiosyncrasies of each tool. We argue that simply teaching agents to run classical workflows, or reimplementing the same tools in faster languages, misses the root of the problem. Instead, we propose a paradigm flip: make genomic data operate natively within modern analytics and ML systems, circumventing many traditional tools. Here, we introduce CompoSeq (https://composeq.dev), an initiative to openly develop specifications and reference implementations of composable primitives for genomic analysis. CompoSeq is led by a team of maintainers of established open-source genomic analysis and ML libraries and is currently focused on four workstreams: transport (bridging genomic formats to in-memory representations native to modern engines and tensor libraries); execution (optimized kernels for spatial joins and rasterization); portability (a SQL dialect for genomic interval operations that transpiles across engines); and modeling (loaders converting record batches into rasterized tensors on GPU for sequence-to-function model training).

Composable foundations for agentic genomics

Conrad Bzura[1], Marek Wiewiórka[2], Vedat Yilmaz[1], Tomasz Gambin[2], Jacob Schreiber[1], Nezar Abdennur[1,3,*]

1. Department of Genomics and Computational Biology, UMass Chan Medical School, Worcester, MA, USA
2. Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland
3. Department of Systems Biology, UMass Chan Medical School, Worcester, MA, USA
**E-mail: nezar.abdennur@umassmed.edu*

Autonomous agents are rapidly becoming credible collaborators in computational biology, but their competence is bounded by the tools and interfaces given to them. For genomic analysis, that tooling is fragmented, brittle, and tightly coupled to specific file formats and line-oriented processing. Consequently, most bioinformatics code is dedicated to parsing and I/O, and most time is wasted routing data through tools that inadvertently create intermediate copies at every step, and handling edge cases/deviations with ad hoc scripts. These influences not only bottleneck traditional analyses but also increasingly limit the scalability of training genomic machine learning (ML) models and usability by agents, who, like humans, must battle with the idiosyncrasies of each tool. We argue that simply teaching agents to run classical workflows or reimplementing the same tools in faster languages does not address the root of the problem. Rather, we propose to flip the paradigm by making genomic data operate natively within modern analytics and ML systems, circumventing the need for many traditional tools. Put another way, with the right primitives, agents can reason more deeply about biology than about file formats. As maintainers of established open-source libraries for genomic analysis and ML, we have begun a joint initiative called CompoSeq (https://composeq.dev) to develop the required specifications and reference implementations based on open, domain-agnostic standards, with a focus on the following workstreams:

1. Transport: Bridging genomic formats to efficient interchange and in-memory representations that are native to modern analytic engines and tensor libraries.
2. Execution: Optimized reference kernels for the operations that dominate genomic analysis: spatial joins (overlap, nearest-feature, and related operations) and rasterization (coverage depth, pileup, windowed statistics).
3. Portability: A SQL dialect for genomic interval operations that transpiles to standard SQL and other engines (DuckDB, DataFusion, Polars, as well as warehouse/lakehouse platforms), so that queries outlive the engine they are written against.
4. Modeling. Data loaders that efficiently convert tabular record batches into rasterized tensors (PyTorch, JAX, Tensorflow, etc.) on the GPU for sequence-to-function model training.

We will introduce the projects, architecture, concepts, and standards of the CompoSeq initiative and report on early progress and benchmarks. By decoupling genomic query semantics from data transport and query execution, our goal is a fabric for genomic data that is rapid, declarative, streaming, and composable, making it an ideal substrate for agent-driven genomic research. Whether fine-mapping variants or training machine learning models, the composable foundations outlined in this talk provide the expressiveness and the power to automatically optimize queries (via push-down, projection, and pruning), migrate to cloud-native storage, avoid slow and wasteful file generation and bespoke shell scripts, and accommodate arbitrary customization, static query analysis, and verification. They will enable researchers, human and agentic, to operate on genomic data declaratively using standard interfaces (SQL, lazy dataframes, tensors). All specifications, benchmarks, and materials are developed in the open under permissive licenses.