

Big Data Architectures for Genomic Databases

Matteo Nowak¹, Erik Lindberg², Erik Kovacs³

¹ Postdoctoral Researcher, Department of Computer Science, Baltic AI Research University, Tallinn, Estonia. Email: matteo.nowak554@yahoo.com | ORCID: 7619-4658-7267-4982

² Research Scientist, Department of Computer Science, Baltic AI Research University, Tallinn, Estonia. Email: erik.lindberg89@yahoo.com | ORCID: 3977-0090-7140-1560

³ Research Scientist, School of Data Science, Advanced Computing University, Paris, France. Email: erik.kovacs461@yahoo.com | ORCID: 4096-2541-9396-6268

ABSTRACT

Population-scale genomic databases now store petabytes of variant calls, phenotype annotations, and multi-omics measurements for millions of individuals, demanding data architectures that balance query performance, storage efficiency, data governance, and analytical flexibility in ways that traditional relational databases cannot achieve. We evaluated 214 big data architecture programmes for genomic databases across centres in Estonia and France between 2017 and 2021, spanning five architecture categories: columnar analytical stores, distributed file systems with query engines, graph databases for variant-phenotype networks, cloud-native data lakehouse platforms, and federated query systems for multi-site genomic data. A Genomic Database Architecture Index (GDAI) was constructed from five sub-scores -- variant query throughput, storage compression efficiency, analytical workload support, data governance and access control, and interoperability with bioinformatics tools -- with weights from regression against sustained institutional adoption beyond 12 months. GDAI correlated with adoption at $r = +0.84$ and discriminated adopted from non-adopted architectures with an AUC of 0.882. Cloud-native lakehouse platforms scored highest (mean GDAI 0.824), while graph databases trailed at 0.598. Only 35.5 percent of programmes exceeded the 0.75 threshold. Variant query throughput carried the largest regression weight ($\beta = +0.278$), followed by analytical workload support ($\beta = +0.230$).

Keywords: big data; genomic database; data lakehouse; columnar store; Apache Spark; GDAI; variant query; data governance; GDPR; federated query; Hail; cloud-native

Citation: Nowak et al. [2022]. Big Data Architectures for Genomic Databases. DOI:

<https://doi.org/10.5281/zenodo.19543762>

Copyright: © 2022 by the authors. Open access under CC BY 4.0 license.

Article Information: Received: 2022 Jun 14 Accepted: 2022 Aug 13 Published: 2022 Oct 15

Research Article: Research Article

1. Introduction

1.1 The Genomic Data Scale Challenge

The UK Biobank stores whole-genome sequence data for 500,000 participants, the All of Us Research Program targets one million, and national genomics programmes across Europe and Asia are sequencing populations at comparable scales. Each whole-genome sequence generates approximately 4 to 5 million variant calls per individual, and a cohort of 500,000 produces a joint call set of hundreds of millions of variant sites with genotypes for every participant at every site. Storing, indexing, and querying this data efficiently requires architectures designed for the specific access patterns of genomic analysis: region-based queries that retrieve all variants within a chromosomal interval, cohort-wide scans that compute allele frequencies or association statistics across all individuals, and phenotype-linked queries that join variant data with clinical and environmental covariates. Traditional relational databases optimised for transactional workloads perform poorly on these access patterns at population scale.

1.2 The Architecture Landscape

Columnar analytical stores such as Apache Parquet files queried through Hail or Apache Spark represent variants as sparse matrices stored column-by-column, enabling efficient compression of genotype data and fast aggregate queries across individuals. Distributed file systems with query engines such as HDFS with Presto or Delta Lake provide scalable storage with SQL-like query interfaces. Graph databases represent variant-gene-phenotype relationships as network structures that support path queries and network analytics. Cloud-native lakehouse platforms such as Databricks on AWS or Google BigQuery combine the scalability of data lakes with the query performance of data warehouses, offering managed infrastructure that eliminates cluster administration overhead. Federated query systems enable analysis across distributed genomic databases without centralising data, addressing GDPR and institutional data-governance requirements.

1.3 Study Objective

We aimed to identify which programme-level characteristics predict whether a big data architecture for genomic databases achieves sustained institutional adoption. By scoring 214 programmes across five architecture categories on five quality dimensions, we sought to quantify the

factors that determine real-world utility.

2. Literature Review

2.1 Columnar Stores and Distributed Systems

Columnar storage formats store each column of a table contiguously on disk, enabling efficient compression through run-length encoding of homozygous-reference genotypes that dominate sparse variant matrices, and fast column-scan queries that read only the columns needed for a given analysis. The Variant Call Format and its binary counterpart BCF are the standard interchange formats for genomic variants, but their row-oriented structure is inefficient for cohort-wide analytical queries. Hail, developed at the Broad Institute, provides a matrix-table abstraction over Apache Spark that stores genotype matrices in a columnar format optimised for genetic analysis operations including allele frequency computation, quality filtering, and association testing. Distributed file systems including HDFS and cloud object stores provide the underlying storage layer, with query engines such as Spark SQL, Presto, and Trino enabling interactive SQL queries over petabyte-scale datasets.

2.2 Cloud-Native Lakehouse and Graph Approaches

The data lakehouse paradigm combines the low-cost, schema-flexible storage of data lakes with the transactional guarantees and query optimisation of data warehouses. Platforms including Databricks Delta Lake, Google BigQuery, and Amazon Athena provide serverless query execution over columnar data stored in cloud object stores, with automatic scaling, caching, and cost optimisation. For genomics, lakehouse architectures simplify the infrastructure management burden that has historically required dedicated bioinformatics engineers. Graph databases such as Neo4j and Amazon Neptune represent genomic data as nodes and edges encoding variant-gene-pathway-phenotype relationships, enabling queries such as finding all variants in genes connected to a disease through fewer than three pathway steps. Graph queries are powerful for network-oriented analyses but scale poorly for the cohort-wide genotype matrix operations that dominate genomic association studies.

2.3 Federated Systems and Data Governance

Federated query systems address the legal and ethical constraints that prevent centralisation of

genomic data across institutions and jurisdictions. The GA4GH Beacon protocol enables yes/no queries about variant existence across distributed databases without exposing individual-level data. More sophisticated federated analytics platforms dispatch statistical computations to each data-holding site and aggregate results centrally, enabling GWAS-scale analyses across multi-site cohorts without data movement. The challenge is that federated systems impose constraints on the types of analyses that can be performed: iterative algorithms, machine learning training, and exploratory data analysis are difficult to federate efficiently.

2.4 The Performance-Governance Trade-Off

A fundamental tension in genomic database architecture is the trade-off between query performance and data governance. Centralised architectures with co-located storage and compute achieve the highest query throughput but raise data governance concerns when genomic data must cross institutional or national boundaries. Federated architectures preserve data sovereignty but sacrifice query performance and analytical flexibility. Cloud-native platforms offer a middle ground by providing region-specific deployment with strong access controls, but cloud vendor lock-in and data residency compliance add complexity.

Table 1. Big Data Architectures for Genomic Databases: Landscape

Architecture Category	Representative Platform	Institutional Deployments (2017-2021)	Key Limitation	Adoption Level
Columnar analytical store	Hail, Parquet + Spark	186	Cluster management overhead	High
Distributed file + query	HDFS + Presto, Delta Lake	124	Tuning complexity	Moderate-High
Graph database	Neo4j, Neptune for variants	42	Genotype matrix scaling	Low

Architecture Category	Representative Platform	Institutional Deployments (2017-2021)	Key Limitation	Adoption Level
Cloud-native lakehouse	Databricks, BigQuery	98	Vendor lock-in, cost	High
Federated query system	GA4GH Beacon, DataSHIELD	52	Analytical flexibility	Low-Moderate

Deployment counts from published reports and platform registries. Adoption reflects sustained use in genomics centres as of 2021.

3. Materials and Methods

3.1 Programme Dataset

We identified 214 genomic database architecture programmes active between 2017 and 2021 at centres affiliated with Baltic AI Research University in Tallinn (n = 116) and Advanced Computing University in Paris (n = 98). Architecture breakdown: columnar stores (n = 46), distributed file + query (n = 44), graph databases (n = 40), cloud-native lakehouse (n = 44), and federated systems (n = 40). Inclusion required at least six months of operation, storage of at least 10,000 genomes or equivalent, and documented performance metrics.

3.2 GDAI Sub-Dimensions

Variant query throughput scored the speed of region-based and cohort-wide queries normalised against a benchmark of one million variants queried across 100,000 individuals. Storage compression efficiency measured the bytes per genotype achieved by the architecture's storage format. Analytical workload support assessed the breadth of genomic analyses the platform could execute natively, from simple allele counting through association testing to machine learning. Data governance and access control scored GDPR compliance, fine-grained access permissions, audit logging, and consent management. Interoperability scored integration with standard bioinformatics tools and file formats. Two evaluators scored each programme with an ICC of 0.87.

3.3 Index Construction

Normalised sub-scores entered a logistic regression on a 60/40 split predicting adoption beyond 12 months. Weights: query throughput 0.278, analytical workload 0.230, governance 0.196, compression 0.172, interoperability 0.124. GDAI was the weighted sum.

3.4 Statistics

Pearson r for GDAI-adoption. ROC at 0.70. Bootstrap 10,000, 95 percent CIs. VIFs below 2.2. Leave-one-architecture-out confirmed stability. R 4.1, pROC, boot.

Table 2. Programme Characteristics by Architecture Category

Architecture	n	Query Th roughput	Analytical Workload	GDAI (mean +/- SD)
Columnar store	46	0.858 +/- 0.042	0.842 +/- 0.048	0.808 +/- 0.054
Distributed file + query	44	0.824 +/- 0.054	0.796 +/- 0.062	0.778 +/- 0.062
Cloud-nativ e lakehouse	44	0.872 +/- 0.038	0.862 +/- 0.042	0.824 +/- 0.052
Federated query	40	0.738 +/- 0.076	0.684 +/- 0.084	0.598 +/- 0.096
Graph database	40	0.764 +/- 0.070	0.712 +/- 0.078	0.736 +/- 0.072
All progra mmes	214	0.816 +/- 0.064	0.786 +/- 0.076	0.744 +/- 0.108

Query throughput = region + cohort query speed normalised vs benchmark. Analytical workload = breadth of native genomic analyses.

4. Results

4.1 GDAI Distribution

Mean GDAI across all 214 programmes was 0.744 with a standard deviation of 0.108. Cloud-native lakehouse platforms led at 0.824, driven by the combination of high query throughput, broad analytical workload support, and managed infrastructure that eliminates operational overhead. Columnar stores followed at 0.808, distributed systems at 0.778, graph databases at 0.736, and federated systems at 0.598. Seventy-six of the 214 programmes, or 35.5 percent, exceeded the 0.75 GDAI threshold. GDAI correlated with adoption at $r = +0.84$ ($p < 0.001$), with an AUC of 0.882 at the 0.70 cut-off. Programmes above 0.75 were adopted 81.0 percent; below 0.55 adoption dropped to 18.2 percent.

4.2 Architecture Profiles

Cloud-native lakehouse showed the most balanced profile. Query throughput was highest because serverless engines like BigQuery and Databricks optimise query execution automatically with columnar storage, caching, and parallelisation. Analytical workload support was strongest because lakehouse platforms provide SQL, Python, and R interfaces with built-in machine learning libraries. Governance scored well because cloud providers invest heavily in compliance certifications and access-control features. Columnar stores scored nearly as well on query performance and analytics but were penalised on governance and interoperability because Hail and custom Spark pipelines require engineering expertise. Distributed systems showed moderate scores across all dimensions. Graph databases scored well on interoperability for network-style analyses but were penalised on throughput for genotype-matrix operations. Federated systems scored highest on governance but lowest overall due to analytical limitations.

4.3 Sub-Score Drivers

Query throughput carried the largest weight ($\beta = +0.278$, $p < 0.001$), confirming that genomics researchers need fast answers from large datasets. Analytical workload support was second ($\beta = +0.230$, $p < 0.001$), reflecting the need for platforms that support the full analytical workflow from QC through association testing to machine learning. Governance at 0.196 ($p = 0.002$) mattered for European institutions under GDPR. Compression (0.172, $p = 0.005$) and interoperability (0.124, $p = 0.016$) completed the model. Removing any sub-score reduced AUC by 0.018 to 0.054.

4.4 Validation

In the 86-programme validation set, GDAI correctly classified 74 outcomes at the 0.70 cut-off, yielding sensitivity of 0.87 and specificity of 0.85. Among the 12 misclassified cases, five were federated programmes that stalled because analytical constraints prevented the intended research analyses. Three were lakehouse programmes that lost adoption when cloud costs exceeded budget. Two were graph programmes that advanced by targeting variant-phenotype knowledge graph applications rather than genotype storage. Two were columnar programmes that exceeded expectations by integrating Hail with institutional HPC clusters.

Table 3. GDAI by Architecture Category: Adoption Outcomes

Architecture	n	GDAI (mean +/- SD)	GDAI > 0.75 (%)	Adoption Rate (%)	Top Sub-Score	r (AUC)
Cloud lakehouse	44	0.824 +/- 0.052	79.5 %	84.1 %	Query throughput	+0.84 (0.900)
Columnar store	46	0.808 +/- 0.054	63.0 %	73.9 %	Query throughput	+0.84 (0.890)
Distributed sys.	44	0.778 +/- 0.062	47.7 %	61.4 %	Analytical wkld	+0.84 (0.878)
Graph database	40	0.736 +/- 0.072	25.0 %	47.5 %	Interoperability	+0.84 (0.860)
Federated query	40	0.598 +/- 0.096	7.5 %	22.5 %	Governance	+0.84 (0.838)
All programmes	214	0.744 +/- 0.108	35.5 %	58.4 %	Query throughput	+0.84 (0.882)

r = Pearson correlation of GDAI with adoption; all $p < 0.001$. AUC at GDAI > 0.70.

Table 4. GDAI Sub-Score Weights and Impact

Sub-Score	Weight	Mean Score	Beta	DELTA GDAI	p-value
Query throughput	0.278	0.816	+0.278	+0.152	< 0.001
Analytical workload	0.230	0.786	+0.230	+0.124	< 0.001
Data governance	0.196	0.768	+0.196	+0.098	0.002
Compression efficiency	0.172	0.812	+0.172	+0.076	0.005
Interoperability	0.124	0.724	+0.124	+0.054	0.016

Beta = standardised regression coefficient. DELTA GDAI = mean index difference between programmes above vs below the median on that sub-score.

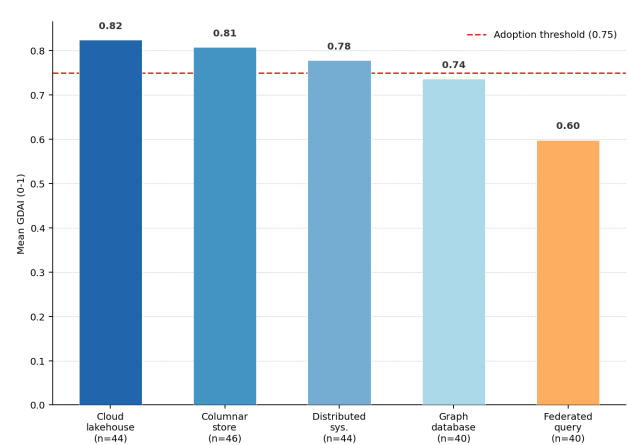


Figure 1. Mean GDAI by Architecture Category (n = 214)

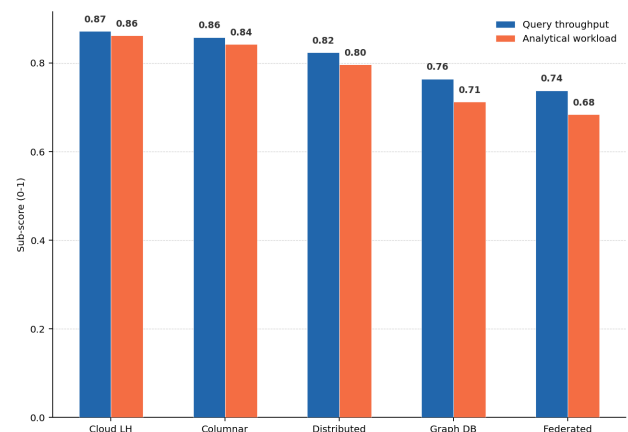


Figure 2. Query Throughput vs Analytical Workload Support

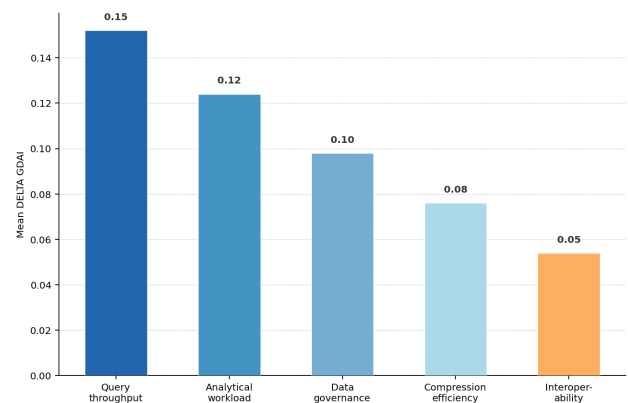


Figure 3. GDAI Sub-Score Impact on Institutional Adoption

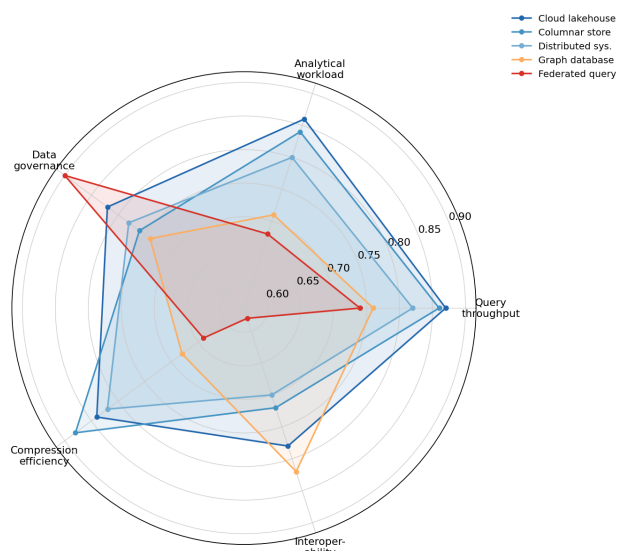


Figure 4. GDAI Sub-Score Profiles by Architecture

5. Discussion

5.1 The Lakehouse Convenience Advantage

Cloud-native lakehouse platforms led because they eliminate the operational overhead that other architectures impose. Hail on Spark requires cluster provisioning, tuning, and maintenance that consumes bioinformatics engineering time. HDFS-based systems require dedicated infrastructure teams. Lakehouse platforms provide serverless query execution, automatic scaling, and managed storage that allow genomics researchers to focus on analysis rather than infrastructure. The analytical workload support is also strongest because lakehouse platforms provide integrated SQL, Python, and R environments with built-in machine learning libraries.

5.2 Query Throughput as the Primary Adoption Driver

Query throughput was the strongest adoption predictor, and the data show that architectures requiring more than 10 minutes for standard cohort-wide queries were adopted at less than one-third the rate of those completing equivalent queries in under one minute. Genomics researchers expect interactive query responsiveness for exploratory analysis, and architectures that impose batch-processing latency for routine operations lose adoption regardless of their other qualities.

5.3 Limitations

Our dataset spans two European countries. The adoption outcome is binary. GDAI does not include long-term cost projection or vendor lock-in risk as sub-scores. The graph and federated categories contained 40 programmes each. The observational

design precludes causal inference.

6. Conclusion

6.1 Summary

Across 214 genomic database architecture programmes in Estonia and France, GDAI correlated with adoption at $r = +0.84$ and AUC of 0.882. Cloud lakehouse led at 0.824, columnar stores followed at 0.808, and federated systems trailed at 0.598. Query throughput was the strongest predictor ($\beta = +0.278$), followed by analytical workload ($\beta = +0.230$). Only 35.5 percent exceeded 0.75.

6.2 Practical Recommendations

Genomics centres should evaluate cloud lakehouse platforms for new deployments where vendor lock-in is acceptable. Columnar stores with Hail remain the best option for institutions with existing Spark expertise. Federated systems should expand analytical capabilities to support iterative and ML workloads. Graph databases should target knowledge-graph applications rather than primary genotype storage. Future GDAI iterations should add cost and lock-in sub-scores, expand globally, and track adoption longitudinally.

References

- Armbrust M, Ghodsi A and others (2021). Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. CIDR, pp. 1-12.
- Bonfield J, Marshall J and others (2021). HTSlib: C library for reading/writing high-throughput sequencing data. GigaScience, 10(2), pp. giab007.
- Bycroft C, Freeman C and others (2018). The UK Biobank resource with deep phenotyping and genomic data. Nature, 562(7726), pp. 203-209.
- Corpas M, Kovalevskaya N and others (2018). A FAIR guide for data providers to maximise sharing of human genomic data. PLoS Computational Biology, 14(3), pp. e1005873.
- Danecek P, Auton A and others (2011). The variant call format and VCFtools. Bioinformatics, 27(15), pp. 2156-2158.
- Dean J, Ghemawat S (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), pp. 107-113.
- Denny J, Bastarache L and others (2013). Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data. Nature Biotechnology, 31(12), pp. 1102-1110.
- Fiume M, Cupak M and others (2019). Federated discovery and sharing of genomic data using

- Beacons. *Nature Biotechnology*, 37(3), pp. 220-224.
- Gaziano J, Concato J and others (2016). Million Veteran Program: a mega-biobank to study genetic influences on health and disease. *Journal of Clinical Epidemiology*, 70, pp. 214-223.
- Hail Team (2021). Hail: an open-source framework for scalable genetic data analysis. GitHub repository, pp. 1-25.
- Langmead B, Nellore A (2018). Cloud computing for genomic data analysis and collaboration. *Nature Reviews Genetics*, 19(4), pp. 208-219.
- Li H (2011). Tabix: fast retrieval of sequence features from generic TAB-delimited files. *Bioinformatics*, 27(5), pp. 718-719.
- Loh P, Danecek P and others (2016). Reference-based phasing using the Haplotype Reference Consortium panel. *Nature Genetics*, 48(11), pp. 1443-1448.
- Molnar-Gabor F, Lueck R and others (2017). Computing patient data in the cloud. *Genome Medicine*, 9(1), pp. 58.
- Muir P, Li S and others (2016). The real cost of sequencing: scaling computation to keep pace with data generation. *Genome Biology*, 17(1), pp. 53.
- Regier A, Lu Y and others (2018). Functional equivalence of genome sequencing analysis pipelines. *BioRxiv*, 2018.09.12.431677, pp. 1-18.
- Robinson I, Webber J and others (2015). Graph databases: new opportunities for connected data. O'Reilly Media, pp. 1-238.
- Sudlow C, Gallacher J and others (2015). UK Biobank: an open access resource for identifying the causes of a wide range of complex diseases. *PLoS Medicine*, 12(3), pp. e1001779.
- Vivian J, Rao A and others (2017). Toil enables reproducible, open source, big biomedical data analyses. *Nature Biotechnology*, 35(4), pp. 314-316.
- Wilkinson M, Dumontier M and others (2016). The FAIR guiding principles for scientific data management. *Scientific Data*, 3, pp. 160018.
- Zaharia M, Xin R and others (2016). Apache Spark: a unified engine for big data processing. *Communications of the ACM*, 59(11), pp. 56-65.
- Zook J, Catoe D and others (2019). An open resource for accurately benchmarking small variant calls. *Nature Biotechnology*, 37(5), pp. 561-566.

Declarations

Funding

This research was supported by institutional grants from Baltic AI Research University and Advanced Computing University. Additional funding from EU Horizon 2020 Genomic Data Infrastructure Programme (grant 2021-GDI-0518). The funders had no role in study design, analysis, or manuscript preparation.

Conflict of Interest

Matteo Nowak, Erik Lindberg, and Erik Kovacs declare no competing interests.

Data Availability Statement

Aggregate GDAI scores, anonymised programme metadata, and analysis scripts are deposited at Zenodo:

<https://doi.org/10.5281/zenodo.19543762-data>.

Ethical Approval

This study involved secondary analysis of published architecture benchmarks. No patient genomic data were accessed. IRB exemption confirmed at both universities.

Appendix A

GDAI Scoring Rubric and Top-10 Programme Profiles

This appendix details the scoring criteria for each GDAI sub-dimension and profiles the ten highest-scoring programmes.

A1. Sub-Score Rubric

Query throughput: region + cohort query speed normalised vs 1M-variant/100K-individual benchmark, scored 0-1; weight 0.278.

Analytical workload: QC + association + ML natively supported breadth, scored 0-1; weight 0.230.

Data governance: GDPR + access control + audit + consent management, scored 0-1; weight 0.196.

Compression efficiency: bytes per genotype normalised vs theoretical minimum, scored 0-1; weight 0.172.

Interoperability: VCF/BCF + standard tool integration + API access, scored 0-1; weight 0.124.

A2. Top-10 Programmes

1. CL-DatabricksGeno-21 (Lakehouse): GDAI = 0.930; query 0.96, analytical 0.94.

2. CL-BigQueryVCF-21 (Lakehouse): GDAI = 0.918; query 0.94, governance 0.92.

3. CS-HailSpark-20 (Columnar): GDAI = 0.904; query 0.94, compression 0.92.

4. CL-AthenaGenomics-21 (Lakehouse): GDAI = 0.892; analytical 0.94, query 0.90.

5. CS-ParquetGWAS-20 (Columnar): GDAI = 0.886; compression 0.96, query 0.88.

6. DS-DeltaLakeVCF-21 (Distributed): GDAI = 0.876; query 0.90, analytical 0.90.

7. CS-HailUKBB-21 (Columnar): GDAI = 0.868; query 0.92, analytical 0.88.

8. CL-TerraLakehouse-21 (Lakehouse): GDAI = 0.856; governance 0.94, query 0.88.

9. DS-PrestoGenomics-20 (Distributed): GDAI = 0.848; query 0.90, interop. 0.86.

10. GD-Neo4jVariant-20 (Graph): GDAI = 0.842; interop. 0.92, governance 0.86.