IntuitionLabs

INTUITIONLABS / RESEARCH & PRACTICAL GUIDANCE

Pharma R&D Data Lakehouses: Databricks, Snowflake & Iceberg

> Build practical AI for pharma and biotech with IntuitionLabs. We help life-science teams turn complex information and workflows into useful software, governed knowledge systems and AI tools.

Published by IntuitionLabs · 2026-04-20 · pharma data lakehouse · life sciences analytics · databricks · snowflake · apache iceberg · data architecture · bioinformatics · clinical data management

Original article: https://intuitionlabs.ai/articles/pharma-data-lakehouse-databricks-snowflake-iceberg



In this article

1. Executive Summary
2. Introduction
3. Data in Pharma R&D: Challenges and Requirements
4. The Data Lakehouse Paradigm
5. Databricks Lakehouse Platform in Life Sciences
6. Snowflake Data Cloud for Life Sciences
7. Apache Iceberg: Open Table Format for Life Sciences
8. Case Studies and Real-World Examples
9. AstraZeneca: Unified AI Drug Discovery (Databricks)
10. GE Healthcare: Breaking Patient-Care Silos (Databricks)
11. Pfizer: Faster Insights with Snowflake
12. Illumina: Embracing Iceberg in Manufacturing (Snowflake)
13. Academic & Consortium Platforms
14. Implications and Future Directions
15. Conclusion

Executive Summary

Pharmaceutical R&D is generating an unprecedented volume and variety of data, from genomics and imaging to clinical trials and real-world evidence (pmc.ncbi.nlm.nih.gov) (www.techradar.com). Data lakehouses have emerged as a unified architecture to address these challenges by merging the best of cloud data lakes (cheap object storage, schema flexibility) with data warehouses (governance, ACID transactions). Leader platforms such as **Databricks** (with its Spark-based Lakehouse and Delta Lake), **Snowflake** (as a cloud Data Cloud with new support for open formats), and open-table formats like **Apache Iceberg** are at the forefront of this trend. In life sciences, these solutions enable integrated analytics and AI on genomics, clinical, and operational data, while supporting regulatory compliance (FAIR principles, traceability) and collaboration.

This report provides an in-depth comparison of these platforms in the context of Life Sciences R&D. It details architectural principles (e.g. Delta Lake on Databricks, Snowflake's multi-cluster compute, Iceberg's manifest-based tables), specific features (ACID transactions, time-travel, data governance), and real-world use cases. For example, **AstraZeneca** used Databricks to unify "hundreds of sources and millions of data points" into end-to-end AI pipelines (www.casestudies.com); **Illumina** uses Snowflake on Iceberg to analyze "vast datasets" from its manufacturing and genomics instruments (www.snowflake.com); Snowflake's **Pfizer** customer story reports a more than fourfold speedup for a specific analytics process and 57% lower total cost of ownership versus its previous solution (www.snowflake.com). Key differences include Databricks' Spark-centric flexibility vs. Snowflake's fully managed simplicity, and Iceberg's open interoperability vs closed systems. Future trends point to greater multi-engine interoperability (e.g. Snowflake's new Iceberg support (www.snowflake.com)), integration of AI/ML (notebooks with bioinformatics libraries (pmc.ncbi.nlm.nih.gov)), and cloud-scale computing (e.g. Snowflake's GPU containers (pmc.ncbi.nlm.nih.gov) or Databricks' parallel Spark clusters). We conclude that a Lakehouse architecture combining these tools allows life sciences

organizations to store, manage, and analyze diverse data (from raw NGS files to RDF knowledge graphs) in one platform, dramatically accelerating research workflows while maintaining data integrity (arxiv.org) (pmc.ncbi.nlm.nih.gov).

Introduction

The life sciences and pharmaceutical industry is undergoing a **data revolution**. Advances in technologies like next-generation sequencing, high-resolution imaging, electronic health records, IoT sensors in manufacturing, and automated screening have caused data volumes to **explode**. For example, the NCBI Sequence Read Archive grew from just 47.04 GB in 2007 to about 27.93 PB by early 2024 – a ~620,000× increase (pmc.ncbi.nlm.nih.gov). Sequencing one human genome now produces over 200 GB of raw data (www.techradar.com). These vast and heterogeneous datasets offer unprecedented scientific insights but only if they can be efficiently managed and analyzed. As TechRadar observes, life sciences "vast datasets are generated every day from experiments, clinical records, and screening programs," but turning this **"information overload into insight"** is a major challenge (www.techradar.com). Traditional on-premises data warehouses and file systems cannot scale with this growth: *"the scale and speed of data generation in biopharma R&D demand flexible, high-performance infrastructure; traditional on-premise systems struggle to keep up"* (www.techradar.com). Meanwhile, regulatory and collaboration requirements (FAIR data principles, 21 CFR Part 11 compliance, data sharing) further complicate the landscape.

To cope, many organizations have historically built **two-tier architectures** (a cloud data lake for raw ingestion and a separate cloud data warehouse for analytics). However, this approach leads to data duplication, latency, and complexity (arxiv.org). The **data lakehouse** paradigm has emerged as a solution: it combines the openness and scalability of data lakes with the structure and performance of data warehouses (arxiv.org). In a lakehouse, raw data can be ingested once into a cloud object store (e.g. S3, ADLS), and then queried or transformed in place by multiple compute engines, while a metadata layer (table format) provides ACID transactions, schema enforcement, indexing, and other "warehouse-like" features (arxiv.org). This avoids unnecessary data copies ("no-copy" access) and allows seamless scaling to petabytes (arxiv.org). In practice, three major technologies dominate modern lakehouse solutions in life sciences:

- **Databricks Lakehouse**: An enterprise analytics platform built on Apache Spark, with *Delta Lake* as its open-table format. It provides unified compute (Spark clusters) and collaborative workspaces (notebooks) for scientists and data engineers.
- **Snowflake Data Cloud**: A fully-managed cloud data platform originally a data warehouse but now embracing lakehouse principles (supporting semi-structured data and open formats like Iceberg). It decouples storage and compute, offering scalable SQL analytics, strong data sharing, and managed services.
- **Apache Iceberg**: An open `Apache` table format that can run on any compute engine (Spark, Flink, Trino, Snowflake, etc.). It provides ACID transactions, versioning (time travel), and fine-grained metadata on top of Parquet/ORC files, enabling true open lakehouses without vendor lock-in (www.dremio.com).

In this report, we analyze how such lakehouse architectures are applied to Life Sciences R&D. We review the **historical context**, explain the core **lakehouse concepts and requirements**, then dive into each of the above technologies and how they address domain-specific needs (e.g. genomics, cheminformatics, clinical data). We compare technical features and performance of Databricks, Snowflake, and Iceberg-based solutions, with tables summarizing capabilities. We present **case studies** (e.g. AstraZeneca, Pfizer, Illumina, GE Healthcare) to illustrate real-world benefits. Finally, we discuss the implications for data-driven drug discovery, regulatory compliance, and future directions (AI/ML integration, multi-cloud, open science). Throughout, all claims and statistics are backed by current research and industry sources.

Data in Pharma R&D: Challenges and Requirements

Pharmaceutical R&D involves diverse data sources:

- **Experimental & Preclinical Data**: High-throughput screens (e.g. small-molecule assays), genomics (DNA/RNA sequencing, Crispr screens), proteomics, metabolomics, microscopy and imaging data, and more. These are often large binary files (BAM/FASTQ images), as well as derived structured data (variants, expression tables).
- **Clinical Data**: Patient electronic health records (EHRs), clinical trial case report forms (CRFs), lab reports, imaging (radiology scans), patient-reported outcomes. This data can be messy, semi-structured (HL7/FHIR, JSON), and must remain private and compliant with regulations (HIPAA, GDPR).
- **Manufacturing & IoT Data**: Process controls, sensor logs, quality-control metrics from drug manufacturing lines; supply chain and logistics data. Often time-series and machine-generated.
- **Commercial/Real-World Data**: Market data, claims databases, literature (text mining), and other “big data” used for pharmacovigilance or epidemiology.
- **Derived Data & Models**: Machine learning features, simulation outputs (molecular dynamics, docking), and knowledge graphs linking compounds to targets, pathways, etc.

These disparate data types bring specific challenges:

- **Volume & Velocity**: Illumina quotes $200 USD per genome for sequencing consumables on the NovaSeq X Series at list price using the 25B flow cell, assuming 100 Gb per genome (Illumina cost assumptions). Imaging and sensor data stream continuously. Low-latency processing (e.g. near-real-time proteomics or clinical alerting) may also be needed.
- **Variety & Schema Evolution**: Data arrives in many formats (structured tables, CSVs, Parquet, JSON, documents, images). Rigid schemas (as in traditional warehouses) cannot easily accommodate evolving scientific protocols or new types of data without frequent redesign. Lakehouses allow **schema-on-read/write** flexibility (arxiv.org).
- **Data Integration & Duplication**: Data fragmentation remains a challenge: the National Cancer Institute notes that cancer data can be fragmented and compartmentalized (NCI overview). Integrating clinical, genomic, and assay data requires overcoming different standards and eliminating silos. Traditional ETL-

based warehouses lead to data copies and delays. Lakehouse architectures strive for *“no-copy”* federated queries directly from source storage (arxiv.org).

- **Governance & Compliance**: Regulated industries need rigorous data governance. Data versions and lineage must be tracked (audit trails, time-travel queries) so analyses are reproducible. Sensitive data must be secured at rest and in transit, and shared only with proper controls. Lakehouses support these via catalogs and ACID guarantees (see below).
- **Performance for Analytics/AI**: Bioinformatics pipelines and ML training tasks are compute-intensive. A unified platform must support both SQL BI queries and distributed ML (Spark, TensorFlow, R, etc.). Historically, genomic analysis ran on HPC clusters (NVIDIA Clara, Parabricks, etc (blogs.nvidia.com)). Modern architectures aim to push more of this into the cloud lakehouse (e.g. GPU clusters within the lakehouse (pmc.ncbi.nlm.nih.gov)).

In summary, pharmaceutical R&D data needs **scalable storage**, **flexible schemas**, **strong metadata and cataloging**, **transactional consistency**, and the ability to run **heterogeneous compute** (SQL, Python, R, ML frameworks) on a **single unified platform**. A data lakehouse is designed to meet these requirements (www.techradar.com) (arxiv.org).

The Data Lakehouse Paradigm

A **data lakehouse** unifies data warehousing and data lake concepts. Mazumdar *et al.* (2023) define a lakehouse as an architecture that “combines the desirable attributes of an RDBMS-OLAP (data warehousing) and a data lake” (arxiv.org). Key characteristics include:

- **ACID Transactions**: Like a warehouse, it provides reliable ACID support on data lake storage. Each insert/update/delete is transactional and consistent. This “ensures safe concurrent reads and writes” (arxiv.org) even though the data sits in files. For example, Databricks’ Delta Lake brings serializable transaction logs on Parquet files (pages.databricks.com); Snowflake’s engine by default enforces transaction isolation.
- **Open File Formats**: Data is stored in open, columnar storage (e.g. Apache Parquet/ORC). Table formats (Databricks’ Delta, Apache Iceberg, Apache Hudi) add metadata layers on these files. As Mazumdar *et al.* note, a lakehouse is “built on open file formats” like Parquet and open table formats like Iceberg (arxiv.org). This avoids proprietary lock-in and allows multiple engines to read the same data.
- **Unified Storage & Compute**: Storage (cheap object storage) is decoupled from compute clusters. Any compute engine (Spark, Flink, Presto, Snowflake SQL, etc.) can operate on the same underlying data without ETL. This *“separated storage and compute”* model is explicitly highlighted as the foundation of scalability: the lakehouse “takes advantage of low-cost storage… allowing data of any type… in open formats” (arxiv.org).
- **Schema Enforcement with Evolution**: Lakehouses enforce a defined schema on write (like warehouses) to ensure data quality, but they also allow schema evolution without full rewrites (arxiv.org). New columns or data types can be tolerated and merged over time.
- **Time Travel & Versioning**: Advanced table formats track versions (snapshots) of data. Users can query historical versions or rollback. This is crucial for reproducibility in research.

- **Performance Optimizations**: Like warehouses, lakehouses may include indexing, data compaction, caching and query accelerations. Databricks automatically optimizes Parquet files (Delta OPTIMIZE), while Snowflake transparently caches data in SSD-backed layers.

These features mitigate limitations of pure warehouses or lakes. For example, a traditional data lake might have no transactions (requiring lengthy ETL pipelines and risking inconsistency) and proprietary warehouses can't handle raw unstructured data. A lakehouse offers the *"best of both worlds"*: it avoids unnecessary data copies and silos, supports advanced analytics (including ML) directly on raw data, yet yields strong consistency and governance (arxiv.org).

Table 1 contrasts classic warehousing, classic data lakes, and the lakehouse paradigm in terms of key properties. It also highlights how Databricks Delta, Snowflake, and Iceberg implement these aspects.

Architecture	Data Warehouse	Data Lake	Data Lakehouse
Data Types	Structured only	Structured + unstructured (raw files) (www.montecarlodata.com)	All types: raw and structured, with schema enforcement (arxiv.org)
Storage	Proprietary, expensive (append-only blocks)	Cheap object storage (S3/ADLS) (arxiv.org)	Cheap object storage + metadata layer (arxiv.org)
Schema Handling	Schema-on-write	Schema-on-read (raw)	Enforced schema on write + evolving schema support (arxiv.org)
ACID Transactions	Yes (arxiv.org)	No (eventual/none)	Yes (Delta, Iceberg, Snowflake all provide ACID) (arxiv.org)
Compute Engines	Vendor-specific SQL engine (e.g. Snowflake's)	Any (Spark, Presto, Flink, etc.)	Any. Table formats like Iceberg allow multiple engines (www.dremio.com). Databricks uses Spark, Snowflake uses its own engine, etc.
Time Travel	Varies (some support history queries)	Not natively	Yes (table versions)
Performance	Optimized for BI/OLAP (indexes, caches)	Poor (no indexes)	Optimized (indexes, caching, auto-optimizers)
Data Copies	Typically one curated copy (ETL'd)	Many uncurated copies	*No-copy* access: compute reads from source directly (arxiv.org)
Governance/Catalog	Centralized (managed by warehouse)	Ad-hoc (files lack catalog)	Centralized catalog (e.g. Unity Catalog, AWS Glue) above the data

Table 1: Comparison of data warehouse, data lake, and data lakehouse architectures. In a lakehouse, low-cost object storage (e.g. S3/ADLS) is combined with a transaction-capable metadata layer that unifies streaming and batch data processing (pages.databricks.com) (arxiv.org).

Databricks Lakehouse Platform in Life Sciences

Platform Overview. Databricks was founded by the creators of Apache Spark. The Databricks Lakehouse Platform provides managed Spark clusters, a unified analytics runtime, and collaborative notebooks (supporting Python, Scala, R, SQL). Its core storage layer is **Delta Lake**, an open-source table format that adds ACID transactions on Parquet files (pages.databricks.com). As Databricks puts it, “Delta Lake provides ACID transactions, scalable metadata handling, and unifies streaming and batch data processing” (pages.databricks.com). Delta Lake (now governed by the Linux Foundation) ensures reliable SQL updates and time-travel on object storage.

Engine and Compute. Databricks runs on cloud providers (AWS, Azure, GCP) and provisions Spark clusters on demand. Spark’s native support for distributed computing (including GPUs for deep learning) makes it well-suited for heavy bioinformatics and ML tasks. For example, running Spark UDFs with genomics libraries or distributed TensorFlow training can scale to hundreds of machines. Databricks also offers autoscaling and spot/preemptible instances to reduce cost. Within Databricks, data engineers and scientists can use *Delta Sharing* to exchange data, and tools like *Databricks SQL* for BI. The platform’s **Unity Catalog** manages metadata and fine-grained access control across all data and compute.

Delta Lake and Open Formats. Delta Lake tables sit on cloud object storage and are accessed as SQL tables. They are stored in Parquet format (open) but with a Delta transaction log for ACID. This means data remains in open format for interoperability, but transitions like INSERT/UPDATE are atomic. Delta also supports **Change Data Feed** (CDC) and merge operations. Many life sciences teams use Delta Lake to unify genomic and clinical data; for example, genomics VCF and GTF files can be loaded into Delta tables via Spark (using Databricks’ `glow` library or other parsers) for batch processing (medium.com).

Use Cases and Design Patterns. Databricks popularized the *“Medallion Architecture”*: a layered model (Bronze/Silver/Gold tables) where raw data is ingested into **Bronze** (raw), transformed/filtered into **Silver** (cleansed) and aggregated to **Gold** (curated) tables. In pharma R&D, one might ingest raw NGS reads and images into a Bronze layer on S3, perform quality filtering and alignment into Silver Delta tables, and produce variant summary and key clinical features in Gold tables for end-user consumption. All stages can be run on scheduled Spark jobs or Workflows.

Scientific Workloads. Databricks supports a wide range of life science analytics. The Databricks Community posts and documentation show how to process VCFs and other genomics formats: for example, using *Project Glow* on Databricks transforms genomic files into flattened Spark tables (medium.com), enabling SQL queries across cohorts. On real-world projects, Databricks has been used for **clinical trial data modernization**. For instance, Databricks reports that AstraZeneca unified “data spread across hundreds of sources and millions of data points” into a single platform (www.casestudies.com). They built pipelines and knowledge graphs on Databricks that run NLP on biomedical literature and recommend drug targets, all accelerated by the Lakehouse’s parallel Spark processing (www.casestudies.com). Similarly, GE Healthcare’s CTO notes that Databricks enabled them to “unify our data in a single platform with a full suite of analytics and ML capabilities,” eliminating silos in patient care data (en.prnasia.com).

Delta vs. Iceberg. Delta Lake and Apache Iceberg support distinct interoperability paths in Databricks. Iceberg reads through UniForm let Iceberg clients read Delta data by generating Iceberg metadata without rewriting the Parquet files. This requires Databricks Runtime 14.3 LTS or above for writes, a Delta table registered in Unity Catalog, column mapping, and the required reader/writer protocol features (Iceberg reads requirements). Native managed and foreign Iceberg tables instead require a Unity Catalog-enabled workspace and Databricks Runtime 16.4 LTS or above; managed tables additionally require serverless compute. Foreign Iceberg tables are read-only in Databricks (native Iceberg requirements and limitations).

Security and Compliance. Databricks supports end-to-end security: data encryption at rest (in S3/ADLS), encryption in transit, network isolation (VPC or VNet injection), and integration with IAM/SSO. Unity Catalog provides centralized governance, making data discoverable and access-controlled, which aids compliance (helping satisfy FAIR "Accessible" and "Reusable" principles with rich metadata). Auditing and lineage is available via Databricks Compliance features.

Snowflake Data Cloud for Life Sciences

Platform Overview. Snowflake is a SaaS data platform that natively separates storage and compute. It was originally a cloud data warehouse but has evolved into a full **Data Cloud** supporting structured and semi-structured data. Snowflake runs on AWS, Azure, and GCP, and provides a multi-cluster SQL engine with virtually unlimited concurrency. Unlike Databricks (which exposes Spark), Snowflake exposes a SQL interface and a managed data warehouse environment. Snowflake manages the infrastructure; warehouse compute is billed per second after a 60-second minimum whenever compute resources are provisioned (warehouse billing documentation).

Earlier Snowflake mainly targeted BI and analytics, but it has aggressively expanded into life sciences and AI. Snowflake has introduced **Snowpark** (to run Python/Scala workloads), **Snowflake Notebooks**, and **Snowpark Container Services**. It now also embraces open-table formats: as of 2025 Snowflake supports Querying **Apache Iceberg** tables natively (www.snowflake.com), and Snowflake's new Open Data Marketplace curated biomedical datasets (like genomic reference data) are often in Iceberg format.

Key Capabilities. Snowflake offers many features valuable in R&D:

- **Multi-Cloud Flexibility:** Snowflake is identical across cloud providers. Koreeda *et al.* note that Snowflake's high compatibility with AWS/GCP/Azure "allows flexible data management while avoiding the risk of vendor lock-in" (pmc.ncbi.nlm.nih.gov). In practice, this means a pharma company can replicate data across clouds or hub data from different subsidiaries (e.g. a global company using different cloud regions).
- **Zero-Maintenance & Auto-Optimizations:** Snowflake automates tuning. Koreeda *et al.* highlight Snowflake's " [near-]zero maintenance design" which "enables automatic infrastructure optimization, significantly reducing the operational burden" (pmc.ncbi.nlm.nih.gov). Researchers don't have to provision or patch servers; capacity expands automatically.
- **Secure Data Sharing:** Snowflake pioneered live data sharing. Using Snowflake's Secure Data Sharing, one can share datasets between accounts or partners without copying (pmc.ncbi.nlm.nih.gov). For example, clinical data can be shared with CROs or collaborators in other organizations, while still being

governed.

- **ACID & MVCC:** Snowflake guarantees transactional consistency across its data (with time travel of up to e.g. 90 days). This means lab and clinical data loaded into tables can be updated safely while analysts query historical snapshots, facilitating reproducible experiments.
- **Semi-Structured Support:** Snowflake can ingest JSON, Avro, Parquet, ORC directly into VARIANT columns or external tables. This is useful for heterogeneous biomedical data (e.g. storing raw FHIR JSON as VARIANT, or loading Parquet EMR data directly).
- **Python and ML Integration:** Through Snowpark and notebooks, Snowflake now supports Python execution. Built-in packages include popular bioinformatics and ML libraries. For example, Snowflake Notebooks come with **RDKit**, **Biopython**, SciPy, etc., already installed (pmc.ncbi.nlm.nih.gov). This allows chemoinformatics directly in the data platform.

Use Cases in Life Sciences. Snowflake is being used for many life sciences scenarios:

- **Genomics and Variant Analysis:** Snowflake can store and query large genomic datasets. Koreeda *et al.* describe using Snowflake to filter disease-associated variants. They ingested 1000 Genomes VCFs (hundreds of gigabytes) via Snowflake External Stages (pointing to AWS Public Datasets) (pmc.ncbi.nlm.nih.gov). Snowflake treats each VCF as semi-structured data (with compression) and runs SQL queries or UDFs over it. Even tens of millions of variants per genome can be efficiently joined, filtered or aggregated thanks to Snowflake's columnar engine (pmc.ncbi.nlm.nih.gov). Time Travel supports access to historical data within the configured retention period (Snowflake Time Travel documentation). For reproducible analyses, plan to retain the relevant data snapshots, transformation code, and execution context beyond that period when needed.
- **Cheminformatics & Drug Screening:** Snowflake can ingest compound libraries from PubChem/ChEMBL/ZINC into an internal stage (pmc.ncbi.nlm.nih.gov). In a Snowflake Notebook, a scientist can use **RDKit** functions to compute molecular fingerprints or Tanimoto similarity on the fly. For example, Koreeda *et al.* describe a full screening workflow: load ZINC subset, compute similarity to a known inhibitor via RDKit (all in SQL/Python in Snowflake), then visualize and filter results with **Streamlit** (pmc.ncbi.nlm.nih.gov). The entire pipeline (ingest→compute→visualize) runs *within* Snowflake, without exporting data. This dramatically shortens distances between ingestion and insight.
- **Machine Learning & AI:** Snowflake's Snowpark Container Service (SPCS) enables bringing custom Docker images into Snowflake's compute layer (pmc.ncbi.nlm.nih.gov). For instance, one can deploy a TensorFlow or PyTorch container with GPU support in Snowflake. Koreeda *et al.* illustrate using a containerized RStudio/Seurat image for single-cell RNA-seq analysis, and even running molecular dynamics on NVIDIA GPUs (SPCS supports up to 1024 GiB with GPUs) (pmc.ncbi.nlm.nih.gov). In one quote, Illumina (a genomics instruments company) said that running analytics on Apache Iceberg tables in Snowflake "unlocked flexibility and performance in managing [their] manufacturing system data at scale" (www.snowflake.com). In short, Snowflake is evolving into a one-stop platform: Snowpark ML allows training models on the data, and Snowflake integrates with data science ecosystems.
- **Cross-Clinical Data Integration:** Snowflake's Data Cloud is being used to integrate EHR/test data. For example, many healthcare providers use Snowflake to collect HIPAA data lakes – referencing industry standards (HL7/FHIR, OMOP) – to fuel predictive models. (Snowflake's marketing touts partnerships for precision medicine). A platform's storage and catalog capabilities can support implementation of the

FAIR guiding principles, but they do not establish FAIRness by themselves. Implementation also requires persistent identifiers, rich metadata, shared vocabularies, qualified references, clear usage licenses, provenance, and community standards (FAIR Guiding Principles).

- **Enterprise BI & TCO Gains:** Snowflake's Pfizer customer story reports more than 19,000 hours of processing time saved over a year and a 57% reduction in TCO versus the previous solution. A described analytics process improved from about 37 minutes to an average of eight minutes with Snowpark; these are case-specific outcomes (Snowflake customer story). This was achieved by consolidating multiple business units' data into a single Snowflake account, enabling shared analytics and removing the overhead of legacy databases (www.snowflake.com).

Iceberg Support and Open Interoperability. Notably, Snowflake has embraced Apache Iceberg to enhance openness. In April 2025, Snowflake announced full support for Iceberg tables (www.snowflake.com). Snowflake's PR highlights that customers (including Illumina, Komodo Health, Medidata, WHOOP) can now use Iceberg's open tables with Snowflake's engine (www.snowflake.com). As Stephen Horn, Staff Data Solutions Architect at Illumina, states in Snowflake's press release, running analytics on Iceberg in Snowflake gives the "flexibility and performance" to analyze **"vast datasets"** in place (www.snowflake.com). In practical terms, this means Snowflake can directly query S3/ADLS tables in Iceberg format with zero movement; users "gain the best of both worlds: Iceberg's flexibility and Snowflake's powerful platform" (www.snowflake.com). For life sciences collaborations, verify the required access path before choosing an open format: foreign Iceberg tables, for example, are read-only in Databricks (Databricks Iceberg documentation).

Apache Iceberg: Open Table Format for Life Sciences

Overview. Apache Iceberg is an open-source table format (became an Apache project in 2020) designed for huge analytic datasets on data lakes. Iceberg defines tables via JSON/Avro metadata *manifests* that track file paths, partitions, and snapshots. Notably, Iceberg tables are *engine-agnostic*: Spark, Flink, Trino/Presto, Athena, Dremio, and even Snowflake can read/write Iceberg tables (www.dremio.com). According to Dremio's comparison, Iceberg uses a hierarchical manifest-based storage of metadata that "scales to billions of files with fast query planning" (www.dremio.com). It provides ACID guarantees: each transaction (INSERT/UPDATE) writes a new Iceberg manifest snapshot, making operations atomic. Time travel is supported via these snapshots. Schema and partition evolution are built in: columns and partitions can be added without rewriting old data.

Iceberg was created at Netflix (2017) to fix limitations of Hive tables; its core design goal was exactly to enable a "truly open data lakehouse" (www.dremio.com). As Snowflake's blog explains, Iceberg has enabled the "open data lakehouse" by *"decoupling open storage and compute"* (www.snowflake.com). That blog notes the historic tension: traditional warehouses locked data in proprietary formats, and lakes had no ACID or governance. Iceberg and Delta Lake are distinct table formats that layer table semantics over Parquet data files. Delta Lake originated at Databricks (Delta Lake project documentation).

Iceberg vs. Delta Lake (vs. others): Several sources articulate the differences. Dremio (2026) compares Iceberg to Databricks' Delta Lake in detail (www.dremio.com). Key contrasts include:

- **Metadata/Transaction Log**: Iceberg uses an optimized AVRO-based manifest approach; Delta uses a JSON "_delta_log" directory with Parquet checkpoint files. Manifests allow Iceberg to progress to petabyte+ scales more efficiently (www.dremio.com).
- **Engine Compatibility**: Iceberg is *natively* supported by multiple engines (Spark, Flink, Presto/Trino, Athena, Snowflake, etc.) (www.dremio.com). Delta Spark is one implementation of Delta Lake; delta-rs provides Rust and Python APIs without a Spark, Java, or Databricks dependency (delta-rs documentation). Compatibility depends on the implementation and supported table features: enabling UniForm Iceberg reads, for example, restricts writers to clients that support its write-protocol feature (UniForm compatibility requirements).
- **Catalog Integration**: Iceberg has a pluggable catalog API (supporting Hive, AWS Glue, Databricks Unity Catalog, Apache Nessie, Polaris, etc.) (www.dremio.com). Delta Lake typically uses Databricks' Unity Catalog or Hive Metastore.
- **Governance Model**: Iceberg is governed by ASF (community-driven, many vendors contributing); Delta is under the Linux Foundation but steered mostly by Databricks (www.dremio.com). This affects the perception of vendor lock-in.
- **Features**: Both support ACID, merge/upsert (via Iceberg "delete files" or Delta's MERGE), time travel, and data skipping. Dremio notes that Iceberg even supports granular deletes at the row level more naturally.

A *high-level comparison table* is provided below (Table 2), summarizing how Databricks (Delta Lake), Snowflake, and Iceberg differ on key attributes.

Feature	Databricks Lakehouse	Snowflake Data Cloud	Apache Iceberg Format
Platform Nature	Unified analytics platform on Spark	Fully-managed SQL Data Warehouse/Cloud Platform	Open table format (engine-agnostic)
Storage	Cloud object storage (S3/ADLS) with Delta atop (arxiv.org)	Managed cloud storage (stores data in creator's cloud)	Cloud object storage with Iceberg metadata (arxiv.org)
Compute Engine	Apache Spark (Databricks runtime)	Snowflake's proprietary SQL engine	Any compatible engine (Spark, Flink, Trino, Snowflake, etc.) (www.dremio.com)
File Format	Apache Parquet (via Delta Lake uses Parquet + JSON log) (arxiv.org)	Native tables: internally optimized columnar format; Iceberg tables: open Parquet files (native storage; Iceberg storage)	Apache Parquet (with Iceberg manifest files) (arxiv.org)
Transactionality	ACID via Delta Lake log (arxiv.org)	ACID (multi-cluster MVCC)	ACID via Iceberg manifest snapshots
Time Travel / Versioning	Yes (Delta Lake)	Yes (Snowflake Time Travel)	Yes (tables store snapshots, support time travel)

Feature	Databricks Lakehouse	Snowflake Data Cloud	Apache Iceberg Format
Schema & Partition Evolution	Yes (schema enforcement + evolution) (arxiv.org)	Yes (ALTER TABLE, variant type, etc.)	Yes (columns, partitions can evolve without rewriting)
Metadata/Catalog	Databricks Unity Catalog (managed by DBR)	Snowflake's Data Catalog	Multiple catalogs (Glue, Hive, Nessie, etc.) (www.dremio.com)
Engine Interoperability	Delta Spark and other Delta implementations, subject to feature compatibility	SQL and Snowpark APIs (Python, Java, Scala) execute on Snowflake; Iceberg tables enable compatible external-engine access (Snowpark; Iceberg tables)	Engine-agnostic: built for multi-engine use (www.dremio.com)
Vendor Lock-in	Moderate (Delta is open, but optimized for Spark/DBR)	High (proprietary SQL engine, metadata)	Low (open spec; any vendor/tool can implement)
Key Strengths	Scalable analytics/ML, Spark ecosystem	Ease of use, auto-scaling, data sharing, security	Open interoperability, consistency for lake data

Table 2: Comparison of Databricks Lakehouse (Delta Lake), Snowflake, and Apache Iceberg on key features. Given that open storage and transactionality are critical for a lakehouse (arxiv.org), we see that Iceberg offers full openness and engine flexibility (www.dremio.com), whereas Databricks and Snowflake provide tuned, managed engines at the cost of some vendor dependency.

Case Studies and Real-World Examples

AstraZeneca: Unified AI Drug Discovery (Databricks)

AstraZeneca faced the classic biotech data problem: terabytes of disparate internal and public data (gene expression, pathway databases, literature) existed across hundreds of systems, making it slow to build ML models. Using the Databricks Lakehouse, AZ built scalable data pipelines and a **knowledge graph** to power AI over all these data. A Databricks case report highlights that AZ data scientists could now **"process millions of data points from thousands of sources"** in one unified platform (www.casestudies.com). They implemented a recommendation engine and literature-mining (NLP) entirely on Databricks — running Spark ML jobs and graph queries – thereby *compressing the data-to-decision cycle*. Operationally, this meant faster time-to-insight for novel drug hypotheses, because Raw genomics and clinical trial data in Bronze tables flowed through transformational Spark jobs into Silver feature tables, and Gold-ready analytics tables consumed by ML models.

GE Healthcare: Breaking Patient-Care Silos (Databricks)

GE Healthcare's CTO describes a similar story: "One of the biggest challenges... is building a comprehensive view of the patient. The Databricks Lakehouse is helping [us] with a modern, open and collaborative platform to build patient views across care pathways. By unifying our data... we've diminished costly legacy data silos

and equipped our teams with timely and accurate insights." (en.prnasia.com). This quote (from a 2022 Databricks press release) illustrates how unifying imaging, device, and clinical records into one Lakehouse expedites analytics (e.g. diagnostic AI or outcome modeling). It also emphasizes **open, collaborative** aspects: Databricks' notebooks allowed multiple teams to share code on the same data. Early adopters cited in that press include Regeneron and ThermoFisher, indicating broad life-science interest.

Pfizer: Faster Insights with Snowflake

Pfizer's migration to Snowflake clearly demonstrates costs and performance improvements. A Snowflake customer story reports that Pfizer achieved **4× faster** data processing and a **57% reduction in total cost of ownership (TCO)** simultaneously (www.snowflake.com). By moving critical R&D and business data from legacy databases into Snowflake, Pfizer was able to *"unify business units with greater access to insights and seamless data sharing"* (www.snowflake.com). Snowflake reports more than 19,000 hours of processing time saved over a year, rather than measured staff labor. The story quotes Steven Ring, Director of Enterprise Database Solutions at Pfizer. These vendor-published results describe Pfizer's deployment and specific workloads (Snowflake customer story).

Illumina: Embracing Iceberg in Manufacturing (Snowflake)

Illumina — a genomics instrument manufacturer — provides a concrete example of Iceberg+Snowflake in life sciences. As reported in Snowflake press materials, Illumina said: *"By running analytics on Apache Iceberg tables with Snowflake, we've unlocked flexibility and performance in managing our manufacturing system data at scale. This open architecture allows us to seamlessly analyze vast datasets while maintaining cost efficiency"* (www.snowflake.com). In practice, Illumina likely stores machine logs and process data (which can be massive) in Iceberg on S3, and queries it via Snowflake's engine. The "vital insights" speed up process improvements. The key takeaway is that using an open-format (Iceberg) enabled Illumina to leverage both their preferred compute (Snowflake) and keep data interoperable for the future. Komodo Health (a healthcare analytics firm) gives a similar testimonial about combining their Healthcare Map with open tables on Snowflake (www.snowflake.com).

Academic & Consortium Platforms

Beyond individual companies, governmental and non-profit life sciences initiatives are adopting these technologies. For example, NCI's Cancer Research Data Commons provides access to experimental and clinical datasets, which researchers can explore through online portals or analyze using tools provided through NCI's Cloud Resources (NCI resource guide). Biotech startups, universities, and consortia are likewise building lakehouse stacks. Ardigen, an AI-driven bioinformatics firm, announced a partnership with Databricks *"to transform clinical data management with advanced Lakehouse solutions"* (ardigen.com). Industry analysts predict that lakehouses will underpin future *"smart clinical trials"* and real-world evidence platforms in pharma, enabling rapid AI experimentation on combined clinical-genomic datasets.

Implications and Future Directions

The adoption of lakehouse architectures in life sciences has profound implications:

- **Accelerated Drug Discovery**: By breaking data silos, R&D teams can apply AI/ML across all relevant data (e.g. combining omics with patient data), speeding target identification and hypothesis generation. Databricks and Snowflake both emphasize “AI-driving discovery” in pharma use cases.
- **Regulatory Agility**: Assess which electronic records and signatures fall within 21 CFR Part 11 and the applicable FDA predicate rules. Platform versioning alone does not establish compliance. Use a documented risk assessment to determine validation, audit controls, signature controls, and record-retention measures appropriate to the intended use (FDA guidance).
- **Cost and Operational Efficiency**: Moving from on-prem or siloed apps to cloud lakehouses cuts hardware cost and admin overhead. Snowflake reports a 57% TCO reduction for Pfizer after migration and additional optimization; this is a case-specific result (Pfizer customer story). Operational teams can focus on science, not servers.
- **Open Science and Collaboration**: Apache Iceberg and similar open standards lower barriers for data exchange. Consortia can publish datasets in Iceberg on cloud storage; partners (even with different analytics stacks) can all use it. This portends an ecosystem of interoperable tools in bioinformatics.
- **Convergence of HPC and the Cloud**: Traditional HPC tasks (e.g. molecular dynamics, complex simulations, population genomics) are increasingly moving to cloud-native lakehouse setups. Snowflake’s GPU container service (pmc.ncbi.nlm.nih.gov) and Databricks’ cloud HPC offerings mean even the heaviest bio-compute can be integrated. NVIDIA’s Clara Parabricks (2021) already showed how near-Spark genomics can run on GPUs (blogs.nvidia.com); lakehouses will embed this further.
- **AI and LLM Readiness**: Looking ahead, life sciences is embracing large language models (LLMs) and generative AI for drug design. These models require vast training data and distributed training. A lakehouse holding genome, proteome, and literature data (fully indexed and versioned) is a natural staging ground for such AI. Databricks has launched industry-specific models, and Snowflake offers integrated AI services, suggesting future *AI Data Clouds* for pharma.

In terms of **platform evolution**, Snowflake and Databricks continue to blur lines: Snowflake’s support for Iceberg and connected catalogs (www.dremio.com) points to more heterogeneous, multi-engine deployments. Databricks’ support for Delta sharing and Unity Catalog shows convergence with data marketplace and governance trends. We expect future lakehouse platforms to natively support graph analytics (e.g. SPARQL and knowledge graphs), real-time streaming from IoT devices in drug manufacturing, and stronger ML pipelines (AutoML, model registries).

A multi-engine architecture using object storage, Databricks, and Snowflake is an option, but each access path needs compatible table versions and features, catalog access, permissions, and defined read/write ownership. An open format alone does not establish universal write access or transfer governance policies between platforms. For example, foreign Iceberg tables are read-only in Databricks, and UniForm permits writes only from clients supporting its required table feature (Databricks Iceberg limitations; UniForm writer compatibility).

Conclusion

Life Sciences R&D stands to gain immensely from lakehouse architectures built on platforms like Databricks, Snowflake, and open formats like Apache Iceberg. The convergence of cloud-scale storage with data-warehousing semantics allows pharmaceutical organizations to integrate lab, clinical, and operational data in a single, governed repository. The customer examples describe particular deployments, rather than a general benchmark for lakehouse performance or cost. Snowflake's Pfizer story, for example, reports a 57% TCO reduction after migration and additional optimization (Snowflake customer story).

In summary, a viable Pharma R&D data lakehouse architecture will:

1. **Ingest all raw data once** into cloud object storage (S3/ADLS).
2. **Organize it in open table formats** (Delta Lake or Apache Iceberg) to ensure ACID, schema, and audit features (arxiv.org).
3. **Allow multiple compute engines** to query the same data (Spark workflows, Snowflake SQL, Python/ML notebooks), maximizing flexibility (www.dremio.com).
4. **Securely govern the data** via catalog services and share it (internally or via Snowflake's marketplace) with full lineage.

This unified approach aligns with FAIR data principles (www.nature.com), avoids vendor silos, and optimizes for both batch and real-time use cases. As the industry pushes toward AI-driven drug discovery and precision medicine, these lakehouse platforms will become the **foundational data fabric** of life sciences R&D.

All claims above are supported by industry reports, academic papers, and customer testimonials (pmc.ncbi.nlm.nih.gov) (www.casestudies.com) (www.snowflake.com), ensuring a comprehensive view of the current state and future trajectory of lakehouse adoption in the pharmaceutical domain.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://pmc.ncbi.nlm.nih.gov/articles/PMC11765040
2. https://www.techradar.com/pro/big-data-big-challenge-how-life-sciences-turn-information-overload-into-insight
3. https://www.casestudies.com/company/databricks/case-study/how-ai-is-changing-drug-discovery
4. https://www.snowflake.com/en/news/press-releases/snowflake-unveils-apache-iceberg-innovations
5. https://www.snowflake.com/en/customers/all-customers/case-study/pfizer
6. https://arxiv.org/abs/2310.08697
7. https://www.dremio.com/blog/apache-iceberg-vs-delta-lake
8. https://assets.illumina.com/systems/sequencing-platforms/novaseq-x-plus/applications/broad-sequencing.html
9. https://www.cancer.gov/research/infrastructure/bioinformatics
10. https://blogs.nvidia.com/blog/genomics-parabricks-hpc-ai
11. https://pages.databricks.com/delta-lake-open-source-reliability-for-data-lakes-reg.html
12. https://www.montecarlodata.com/blog-snowflake-and-databricks-summit-comparison

13. https://medium.com/%40nitinaggarwal12/building-a-modern-genomics-lakehouse-with-databricks-and-glow-f7f5ac55deff
14. https://en.prnasia.com/releases/apac/databricks-introduces-lakehouse-for-the-healthcare-and-life-sciences-industries-to-drive-transformation-across-healthcare-ecosystem-354014.shtml
15. https://docs.databricks.com/aws/en/delta/iceberg-reads
16. https://docs.databricks.com/aws/en/iceberg
17. https://docs.snowflake.com/en/user-guide/warehouses-considerations
18. https://docs.snowflake.com/en/user-guide/data-time-travel
19. https://www.nature.com/articles/sdata201618
20. https://www.snowflake.com/en/blog/apache-iceberg-data-lakehouse-architecture
21. https://delta-io.github.io/delta-rs/1.5.1/about/contributing
22. https://docs.snowflake.com/en/user-guide/intro-key-concepts
23. https://docs.snowflake.com/en/user-guide/tables-iceberg
24. https://docs.snowflake.com/en/developer-guide/snowpark/index?lang=en
25. https://www.cancer.gov/about-nci/organization/cbiit/news-events/blog/2024/quick-start-guide-cancer-data-science-clinical-oncology
26. https://ardigen.com/data-lakehouses-a-strategic-imperative-for-the-future-of-clinical-studies
27. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application

THE PUBLISHER

About IntuitionLabs

IntuitionLabs is an AI consulting, custom software development and data engineering firm serving pharmaceutical, biotechnology, medical-device and other life-science organizations. We work with clinical, regulatory, medical-affairs, commercial, quality and IT teams to connect technology decisions with the work people need to accomplish.

AI consulting and adoption

Our AI enablement services cover readiness assessments, use-case selection, governance and policies, team workshops, adoption measurement and ongoing advisory support. We help organizations structure the information layer behind AI: source material, context, permissions and maintained knowledge that make generated answers useful and reviewable. Private LLM inference and hosted AI options support teams evaluating how to operate AI with appropriate control over their data and infrastructure.

Software, data and life-science workflows

IntuitionLabs develops custom software for pharma and biotech, integrates enterprise systems, and builds data engineering and business intelligence solutions. Areas of focus include AI agents, regulatory research, medical writing, medical affairs, CMC information, competitive intelligence and clinical-document workflows. Our eTMF intelligence work includes cross-system reconciliation and inspection-readiness support.

Enterprise platforms and regulated delivery

We provide Veeva services, application support, managed services, integrations and custom applications, alongside enterprise content work involving platforms such as Egnyte. For regulated workflows, our services include GxP enablement, computer-system validation and software development addressing 21 CFR Part 11 requirements. The applicable controls, validation responsibilities and acceptance criteria are defined for each engagement.

Work with IntuitionLabs

Explore AI enablement, pharma and biotech software development, data engineering and BI, and Veeva services. Contact IntuitionLabs to discuss your workflow, information sources and implementation needs.

IntuitionLabs publishes educational research to help life-science teams make informed technology decisions. Coverage of a product or organization does not imply a client relationship, endorsement or partnership.

Website: https://intuitionlabs.ai

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.