

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

4/19/2026 • 30 min read

pharma data lakehouse

life sciences analytics

databricks

snowflake

apache iceberg

data architecture

bioinformatics

clinical data management



## 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](#) <sup>(1)</sup> [pmc.ncbi.nlm.nih.gov](https://pubmed.ncbi.nlm.nih.gov/)) <sup>(2)</sup> [www.techradar.com](https://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 <sup>(3)</sup> [www.casestudies.com](https://www.casestudies.com)); **Illumina** uses Snowflake on Iceberg to analyze "vast datasets" from its manufacturing and genomics instruments <sup>(4)</sup> [www.snowflake.com](https://www.snowflake.com)); **Pfizer** reports a 4x speed-up in queries and 57% lower total cost of ownership after migrating to Snowflake <sup>(5)</sup> [www.snowflake.com](https://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 <sup>(6)</sup> [www.snowflake.com](https://www.snowflake.com))), integration of AI/ML (notebooks with [bioinformatics](#) libraries <sup>(7)</sup> [pmc.ncbi.nlm.nih.gov](https://pubmed.ncbi.nlm.nih.gov/))), and cloud-scale computing (e.g. Snowflake's GPU containers <sup>(8)</sup> [pmc.ncbi.nlm.nih.gov](https://pubmed.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 <sup>(9)</sup> [arxiv.org](https://arxiv.org)) <sup>(10)</sup> [pmc.ncbi.nlm.nih.gov](https://pubmed.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 ~620,000x increase <sup>(1)</sup> [pmc.ncbi.nlm.nih.gov](https://pubmed.ncbi.nlm.nih.gov/)). Sequencing one human genome now produces over 200 GB of raw data <sup>(2)</sup> [www.techradar.com](https://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 <sup>(2)</sup> [www.techradar.com](https://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*" <sup>(11)</sup> [www.techradar.com](https://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 <sup>(12)</sup> [arxiv.org](https://arxiv.org)) <sup>(13)</sup> [arxiv.org](https://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 <sup>(12)</sup> [arxiv.org](https://arxiv.org)) <sup>(9)</sup> [arxiv.org](https://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 <sup>(9)</sup> [arxiv.org](https://arxiv.org)) <sup>(14)</sup> [arxiv.org](https://arxiv.org)). This avoids unnecessary data

copies (“no-copy” access) and allows seamless scaling to petabytes (<sup>[14]</sup> [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 (<sup>[15]</sup> [www.dremio.com](#)) (<sup>[16]</sup> [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:** As noted, genomic sequencing costs have plummeted (from \$1000/genome in 2014 to ~\$600 today (<sup>[1]</sup> [pmc.ncbi.nlm.nih.gov](#))), so cohort sizes routinely reach 100K+ individuals. 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 (<sup>[17]</sup> [arxiv.org](#)).

- **Data Integration & Duplication:** Life sciences orgs often have *many systems* (one survey notes “175+ *distinct data systems*” in a top-20 pharma <sup>(18)</sup> sakaradigital.com)). 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 <sup>(19)</sup> 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 <sup>(20)</sup> blogs.nvidia.com)). Modern architectures aim to push more of this into the cloud lakehouse (e.g. GPU clusters within the lakehouse <sup>(8)</sup> 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 <sup>(11)</sup> www.techradar.com <sup>(9)</sup> 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” <sup>(9)</sup> 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” <sup>(9)</sup> arxiv.org) even though the data sits in files. For example, Databricks’ Delta Lake brings serializable transaction logs on Parquet files <sup>(21)</sup> 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 <sup>(22)</sup> 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” <sup>(14)</sup> 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 <sup>(17)</sup> 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 ([23] arxiv.org) ([9] 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) ([24] www.montecarlodata.com)	All types: raw and structured, with schema enforcement ([17] arxiv.org)
Storage	Proprietary, expensive (append-only blocks)	Cheap object storage (S3/ADLS) ([14] arxiv.org)	Cheap object storage + metadata layer ([14] arxiv.org)
Schema Handling	Schema-on-write	Schema-on-read (raw)	Enforced schema on write + evolving schema support ([17] arxiv.org)
ACID Transactions	Yes ([23] arxiv.org)	No (eventual/none)	Yes (Delta, Iceberg, Snowflake all provide ACID) ([9] arxiv.org)
Compute Engines	Vendor-specific SQL engine (e.g. Snowflake's)	Any (Spark, Presto, Flink, etc.) STA	Any. Table formats like Iceberg allow multiple engines ([25] 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 ([19] 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 ([21] pages.databricks.com) ([14] 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 ([21] pages.databricks.com). As Databricks puts it, "Delta Lake provides ACID transactions, scalable metadata handling, and unifies streaming and batch data processing" ([21] 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' `g1ow` library or other parsers) for batch processing (<sup>[26]</sup> [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 (<sup>[26]</sup> [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 (<sup>[3]</sup> [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 (<sup>[3]</sup> [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 (<sup>[27]</sup> [en.pnasia.com](#)).

**Delta vs. Iceberg.** By default Databricks uses Delta Lake tables. However, with newer runtimes (Databricks Runtime 14.3+), Databricks also supports the open **Apache Iceberg** table format (including reading ICEBERG tables and reading Delta via an Iceberg API (<sup>[28]</sup> [docs.databricks.com](#))). This allows Databricks to interoperate more with multi-engine ecosystems. Some organizations may choose Iceberg on Databricks for maximum portability. Key differences: Delta Lake is deeply optimized for Spark workloads, while Iceberg's manifest architecture is more general (we compare below).

**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. It automatically scales storage and compute, provides micro-second elastic pricing, and handles all infrastructure.

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 (<sup>[6]</sup> [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” (<sup>[29]</sup> [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" (<sup>[30]</sup> [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 (<sup>[31]</sup> [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 (<sup>[32]</sup> [pmc.ncbi.nlm.nih.gov](#)). This allows cheminformatics 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) (<sup>[33]</sup> [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 (<sup>[33]</sup> [pmc.ncbi.nlm.nih.gov](#)). Time-travel also ensures any transformation (e.g. normalization or annotation) is reproducible.
- **Cheminformatics & Drug Screening:** Snowflake can ingest compound libraries from PubChem/ChEMBL/ZINC into an internal stage (<sup>[7]</sup> [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** (<sup>[34]</sup> [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 (<sup>[35]</sup> [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) (<sup>[8]</sup> [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" (<sup>[4]</sup> [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). Snowflake's ability to handle both structured (e.g. clinical tables) and unstructured (e.g. whole-genome VARIANTS) data together satisfies FAIR requirements: it supports *Findable* metadata (through the catalog) and *Interoperable* formats (e.g. Parquet or Iceberg) (<sup>[36]</sup> [www.nature.com](#)) (<sup>[37]</sup> [www.snowflake.com](#)).
- **Enterprise BI & TCO Gains:** According to Snowflake customer stories (e.g. Pfizer), migrating disparate pharma data into Snowflake yielded dramatic productivity gains. Pfizer reports saving "19K annual hours" and cutting TCO by 57%

while running queries **4x faster** after moving to Snowflake (<sup>[5]</sup> [www.snowflake.com](http://www.snowflake.com)). This was achieved by consolidating multiple business units' data into a single Snowflake account, enabling shared analytics and removing the overhead of legacy databases (<sup>[5]</sup> [www.snowflake.com](http://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 (<sup>[6]</sup> [www.snowflake.com](http://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 (<sup>[38]</sup> [www.snowflake.com](http://www.snowflake.com)). As Illumina CEO notes, running analytics on Iceberg in Snowflake gives the "flexibility and performance" to analyze "**vast datasets**" in place (<sup>[4]</sup> [www.snowflake.com](http://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" (<sup>[6]</sup> [www.snowflake.com](http://www.snowflake.com)) (<sup>[37]</sup> [www.snowflake.com](http://www.snowflake.com)). This is particularly important for life sciences consortiums and collaborations, where adopting an open format ensures data can move between Snowflake, Databricks, and other engines.

## 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 (<sup>[25]</sup> [www.dremio.com](http://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" (<sup>[39]</sup> [www.dremio.com](http://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" (<sup>[15]</sup> [www.dremio.com](http://www.dremio.com)). As Snowflake's blog explains, Iceberg has enabled the "open data lakehouse" by "*decoupling open storage and compute*" (<sup>[37]</sup> [www.snowflake.com](http://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 forks like Databricks Delta) solve this by layering table semantics on top of Parquet.

**Iceberg vs. Delta Lake (vs. others):** Several sources articulate the differences. Dremio (2026) compares Iceberg to Databricks' Delta Lake in detail (<sup>[16]</sup> [www.dremio.com](http://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 (<sup>[39]</sup> [www.dremio.com](http://www.dremio.com)).
- **Engine Compatibility:** Iceberg is *natively* supported by multiple engines (Spark, Flink, Presto/Trino, Athena, Snowflake, etc.) (<sup>[25]</sup> [www.dremio.com](http://www.dremio.com)). Delta Lake is tightly integrated with Spark (its transactions and optimizations rely on Spark; other engines must use connectors). This means Iceberg tables can be shared seamlessly across organizations using different tools.
- **Catalog Integration:** Iceberg has a pluggable catalog API (supporting Hive, AWS Glue, Databricks Unity Catalog, Apache Nessie, Polaris, etc.) (<sup>[40]</sup> [www.dremio.com](http://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 (<sup>[41]</sup> [www.dremio.com](http://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
<b>Platform Nature</b>	Unified analytics platform on Spark	Fully-managed SQL Data Warehouse/Cloud Platform	Open table format (engine-agnostic)
<b>Storage</b>	Cloud object storage (S3/ADLS) with Delta atop ([22] arxiv.org)	Managed cloud storage (stores data in creator’s cloud)	Cloud object storage with Iceberg metadata ([22] arxiv.org)
<b>Compute Engine</b>	Apache Spark (Databricks runtime)	Snowflake’s proprietary SQL engine	Any compatible engine (Spark, Flink, Trino, Snowflake, etc.) ([25] www.dremio.com)
<b>File Format</b>	Apache Parquet (via Delta Lake uses Parquet + JSON log) ([22] arxiv.org)	Proprietary internal format (transparent to user)	Apache Parquet (with Iceberg manifest files) ([22] arxiv.org)
<b>Transactionality</b>	ACID via Delta Lake log ([9] arxiv.org)	ACID (multi-cluster MVCC)	ACID via Iceberg manifest snapshots
<b>Time Travel / Versioning</b>	Yes (Delta Lake)	Yes (Snowflake Time Travel)	Yes (tables store snapshots, support time travel)
<b>Schema &amp; Partition Evolution</b>	Yes (schema enforcement + evolution) ([17] arxiv.org)	Yes (ALTER TABLE, variant type, etc.)	Yes (columns, partitions can evolve without rewriting)
<b>Metadata/Catalog</b>	Databricks Unity Catalog (managed by DBR)	Snowflake’s Data Catalog	Multiple catalogs (Glue, Hive, Nessie, etc.) ([40] www.dremio.com)
<b>Engine Interoperability</b>	Spark-native (DBR)	SQL only (with external resilience)	Engine-agnostic: built for multi-engine use ([25] www.dremio.com)
<b>Vendor Lock-in</b>	Moderate (Delta is open, but optimized for Spark/DBR)	High (proprietary SQL engine, metadata)	Low (open spec; any vendor/tool can implement)
<b>Key Strengths</b>	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. In diapers that open storage and transactionality are critical for a lakehouse ([9] arxiv.org) ([14] arxiv.org), we see that Iceberg offers full openness and engine flexibility ([25] 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 ([3] 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." <sup>(27)</sup> [en.pnasia.com](https://en.pnasia.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 **4x faster** data processing and a **57% reduction in total cost of ownership (TCO)** simultaneously <sup>(5)</sup> [www.snowflake.com](https://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" <sup>(5)</sup> [www.snowflake.com](https://www.snowflake.com)). In numerical terms, they saved ~19,000 human-hours per year on queries, and cut database costs by over a quarter <sup>(42)</sup> [www.snowflake.com](https://www.snowflake.com)). While a marketing case study, these figures are backed by Pfizer's CIO and illustrate the tangible ROI of a managed cloud data platform for pharma analytics.

## 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" <sup>(4)</sup> [www.snowflake.com](https://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 <sup>(43)</sup> [www.snowflake.com](https://www.snowflake.com)).

## Academic & Consortium Platforms

Beyond individual companies, governmental and non-profit life sciences initiatives are adopting these technologies. For instance, the National Cancer Institute's Cloud (CDISC Data Warehouse) allows distributed analysis on large cohorts (via Spark/Hive), and NIH's "All of Us" research program has explored cloud platforms. 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" <sup>(44)</sup> [ardigen.com](https://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:** The built-in auditability (time travel, versioning) aligns with regulatory requirements for data provenance. Having all data in one governed environment simplifies reporting and compliance (e.g. 21 CFR Part 11 for electronic records). Catalogs with metadata also aid FAIR data sharing with collaborators.
- **Cost and Operational Efficiency:** Moving from on-prem or siloed apps to cloud lakehouses cuts hardware cost and admin overhead. Snowflake’s Pfizer and Databricks’ AZ examples show huge TCO reductions. 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 (<sup>[8]</sup> [pmc.ncbi.nlm.nih.gov](https://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 (<sup>[20]</sup> [blogs.nvidia.com](https://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 (<sup>[45]</sup> [www.dremio.com](https://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).

Importantly, the lakehouse architecture is also technology-agnostic. Life sciences teams can mix and match: for instance, storing raw data in S3 with Iceberg tables, running ETL on Databricks, performing ad-hoc analysis in Snowflake, and sharing results securely – all without reloading the data. The ICEBERG table format and tools like Delta Sharing facilitate this *multi-engine interoperability*. This flexibility is critical in R&D where specific tasks may need specific tools (e.g. Spark for genomics, SQL for statistical analysis).

## 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. As evidenced by case studies (AstraZeneca, Pfizer, Illumina) and research, these lakehouse systems deliver orders-of-magnitude improvements in data access speed, cost-efficiency, and analytical power (<sup>[3]</sup> [www.casestudies.com](https://www.casestudies.com)) (<sup>[5]</sup> [www.snowflake.com](https://www.snowflake.com)).

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 (<sup>[9]</sup> [arxiv.org](https://arxiv.org)) (<sup>[14]</sup> [arxiv.org](https://arxiv.org)).



- [23] <https://arxiv.org/abs/2310.08697#:~:engin...>
  - [24] <https://www.montecarlodata.com/blog-snowflake-and-databricks-summit-comparison#:~:ln%20...>
  - [25] <https://www.dremio.com/blog/apache-iceberg-vs-delta-lake/#:~:Engin...>
  - [26] <https://medium.com/%40nitinaggarwal12/building-a-modern-genomics-lakehouse-with-databricks-and-glow-f7f5ac55deff#:~:By%20...>
  - [27] <https://en.prnasia.com/releases/apac/databricks-introduces-lakehouse-for-the-healthcare-and-life-sciences-industries-to-drive-transformation-across-healthcare-ecosystem-354014.shtml#:~:Offi...>
  - [28] <https://docs.databricks.com/aws/en/delta/uniform#:~:AWS%2...>
  - [29] <https://pmc.ncbi.nlm.nih.gov/articles/PMC11765040/#:~:This...>
  - [30] <https://pmc.ncbi.nlm.nih.gov/articles/PMC11765040/#:~:Snowf...>
  - [31] <https://pmc.ncbi.nlm.nih.gov/articles/PMC11765040/#:~:analy...>
  - [32] <https://pmc.ncbi.nlm.nih.gov/articles/PMC11765040/#:~:with...>
  - [33] <https://pmc.ncbi.nlm.nih.gov/articles/PMC11765040/#:~:Varia...>
  - [34] <https://pmc.ncbi.nlm.nih.gov/articles/PMC11765040/#:~:for%2...>
  - [35] <https://pmc.ncbi.nlm.nih.gov/articles/PMC11765040/#:~:Snowf...>
  - [36] <https://www.nature.com/articles/sdata201618#:~:match...>
  - [37] <https://www.snowflake.com/en/blog/apache-iceberg-data-lakehouse-architecture/#:~:Now%2...>
  - [38] <https://www.snowflake.com/en/news/press-releases/snowflake-unveils-apache-iceberg-innovations/#:~:data...>
  - [39] <https://www.dremio.com/blog/apache-iceberg-vs-delta-lake/#:~:Metad...>
  - [40] <https://www.dremio.com/blog/apache-iceberg-vs-delta-lake/#:~:Catal...>
  - [41] <https://www.dremio.com/blog/apache-iceberg-vs-delta-lake/#:~:Ecosy...>
  - [42] <https://www.snowflake.com/en/customers/all-customers/case-study/pfizer/#:~:19K%2...>
  - [43] <https://www.snowflake.com/en/news/press-releases/snowflake-unveils-apache-iceberg-innovations/#:~:Illum...>
  - [44] <https://ardigen.com/data-lakehouses-a-strategic-imperative-for-the-future-of-clinical-studies/#:~:Ardig...>
  - [45] <https://www.dremio.com/blog/apache-iceberg-vs-delta-lake/#:~:Engin...>
-

## IntuitionLabs - Industry Leadership & Services

**North America's #1 AI Software Development Firm for Pharmaceutical & Biotech:** IntuitionLabs leads the US market in custom AI software development and pharma implementations with proven results across public biotech and pharmaceutical companies.

**Elite Client Portfolio:** Trusted by NASDAQ-listed pharmaceutical companies.

**Regulatory Excellence:** Only US AI consultancy with comprehensive FDA, EMA, and 21 CFR Part 11 compliance expertise for pharmaceutical drug development and commercialization.

**Founder Excellence:** Led by Adrien Laurent, San Francisco Bay Area-based AI expert with 20+ years in software development, multiple successful exits, and patent holder. Recognized as one of the top AI experts in the USA.

**Custom AI Software Development:** Build tailored pharmaceutical AI applications, custom CRMs, chatbots, and ERP systems with advanced analytics and regulatory compliance capabilities.

**Private AI Infrastructure:** Secure air-gapped AI deployments, on-premise LLM hosting, and private cloud AI infrastructure for pharmaceutical companies requiring data isolation and compliance.

**Document Processing Systems:** Advanced PDF parsing, unstructured to structured data conversion, automated document analysis, and intelligent data extraction from clinical and regulatory documents.

**Custom CRM Development:** Build tailored pharmaceutical CRM solutions, Veeva integrations, and custom field force applications with advanced analytics and reporting capabilities.

**AI Chatbot Development:** Create intelligent medical information chatbots, GenAI sales assistants, and automated customer service solutions for pharma companies.

**Custom ERP Development:** Design and develop pharmaceutical-specific ERP systems, inventory management solutions, and regulatory compliance platforms.

**Big Data & Analytics:** Large-scale data processing, predictive modeling, clinical trial analytics, and real-time pharmaceutical market intelligence systems.

**Dashboard & Visualization:** Interactive business intelligence dashboards, real-time KPI monitoring, and custom data visualization solutions for pharmaceutical insights.

**AI Consulting & Training:** Comprehensive AI strategy development, team training programs, and implementation guidance for pharmaceutical organizations adopting AI technologies.

Contact founder Adrien Laurent and team at <https://intuitionlabs.ai/contact> for a consultation.

---

## DISCLAIMER

The information contained in this document is provided for educational and informational purposes only. We make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability, or availability of the information contained herein.

Any reliance you place on such information is strictly at your own risk. In no event will IntuitionLabs.ai or its representatives be liable for any loss or damage including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising from the use of information presented in this document.

This document may contain content generated with the assistance of artificial intelligence technologies. AI-generated content may contain errors, omissions, or inaccuracies. Readers are advised to independently verify any critical information before acting upon it.

All product names, logos, brands, trademarks, and registered trademarks mentioned in this document are the property of their respective owners. All company, product, and service names used in this document are for identification purposes only. Use of these names, logos, trademarks, and brands does not imply endorsement by the respective trademark holders.

IntuitionLabs.ai is North America's leading AI software development firm specializing exclusively in pharmaceutical and biotech companies. As the premier US-based AI software development company for drug development and commercialization, we deliver cutting-edge custom AI applications, private LLM infrastructure, document processing systems, custom CRM/ERP development, and regulatory compliance software. Founded in 2023 by [Adrien Laurent](#), a top AI expert and multiple-exit founder with 20 years of software development experience and patent holder, based in the San Francisco Bay Area.

This document does not constitute professional or legal advice. For specific guidance related to your business needs, please consult with appropriate qualified professionals.

© 2025 IntuitionLabs.ai. All rights reserved.