

# Scaling Genomic Analyses in Azure

**Presented By:**  
Eric Wozniak  
Colby Ford, Ph.D.



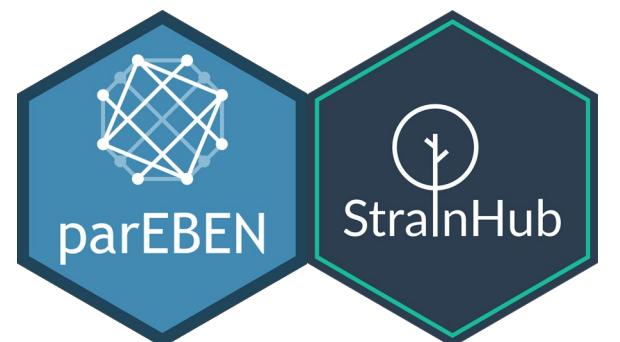
## Eric Wozniak | VP – Industry, Healthcare

- Industry lead for the Healthcare vertical
  - Payors, Providers, Pharmaceuticals, Life Sciences
- Manages strategic accounts by developing go-to-market strategies and offers and building partner relationships.



## Colby Ford | AI Architect

- Focused on Azure cloud architecture for machine learning and data platform use cases
- Ph.D. in Bioinformatics and Computational Biology
- UNC Charlotte researcher
  - Infectious diseases (*E. coli* and *P. falciparum*)
  - Human genomics (Bantu and rare diseases)



**Sparkitecture** The logo features the word "Sparkitecture" in a bold, sans-serif font. To the right of the text is a graphic element consisting of four overlapping colored shapes: red, orange, yellow, and green, forming a stylized letter "S" or a spark.



# Solution Areas

*cloud scale analytics*



## Data Platform

Cloud-based, scalable solutions



## BI & Analytics

Self-service and enterprise



## AI & Machine Learning

Leading edge data science



**BLUE GRANITE**



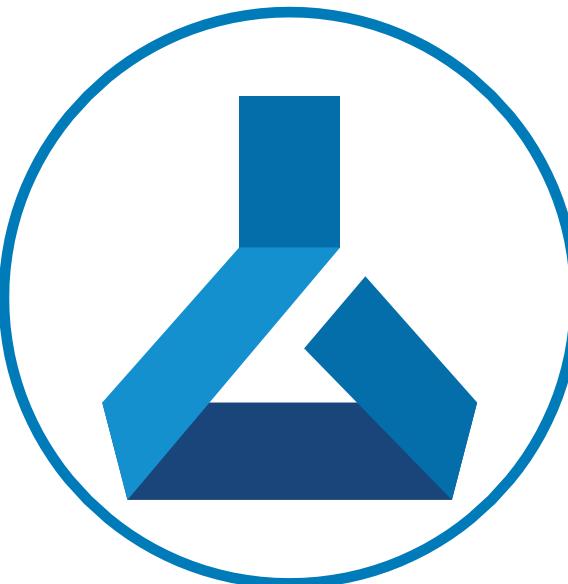
# Azure Services for Genomics-Based Analysis



## Databricks

Apache Spark platform for distributed computing

Includes: Databricks Runtime for Genomics + Glow



## Machine Learning Service

Familiar Jupyter or RStudio environment for interactive analyses and operationalization



## Genomics Service

Automated GATK-Compliant Processing Pipeline



# Azure Databricks

Fast, easy, and collaborative Apache Spark™ based analytics service.

The screenshot shows the Microsoft Azure Databricks portal. On the left is a sidebar with icons for Home, Workspace, Recents, Data, Clusters, Jobs, Models, and Search. The main area has a header for 'SnpEffAnnotationPipeline (Scala)' and a top navigation bar with 'PORTAL' and user info 'cford@blue-granite.com'. Below the header are several tabs: 'Detached', 'File', 'Edit', 'View: Standard', 'Permissions', 'Run All', 'Clear', 'Schedule', 'Comments', 'Runs', and 'Revision history'. A 'replayMode: skip' dropdown is also present. The interface includes three code cells labeled 'Cmd 1', 'Cmd 2', and 'Cmd 3'. 'Cmd 1' contains a pipeline configuration for 'SnpEff Annotation Pipeline'. 'Cmd 2' shows Scala code: 'dbutils.fs.mkdirs("/tmp/genomics")' and 'display(dbutils.fs.ls("/tmp"))'. It also displays a table of files in '/tmp':

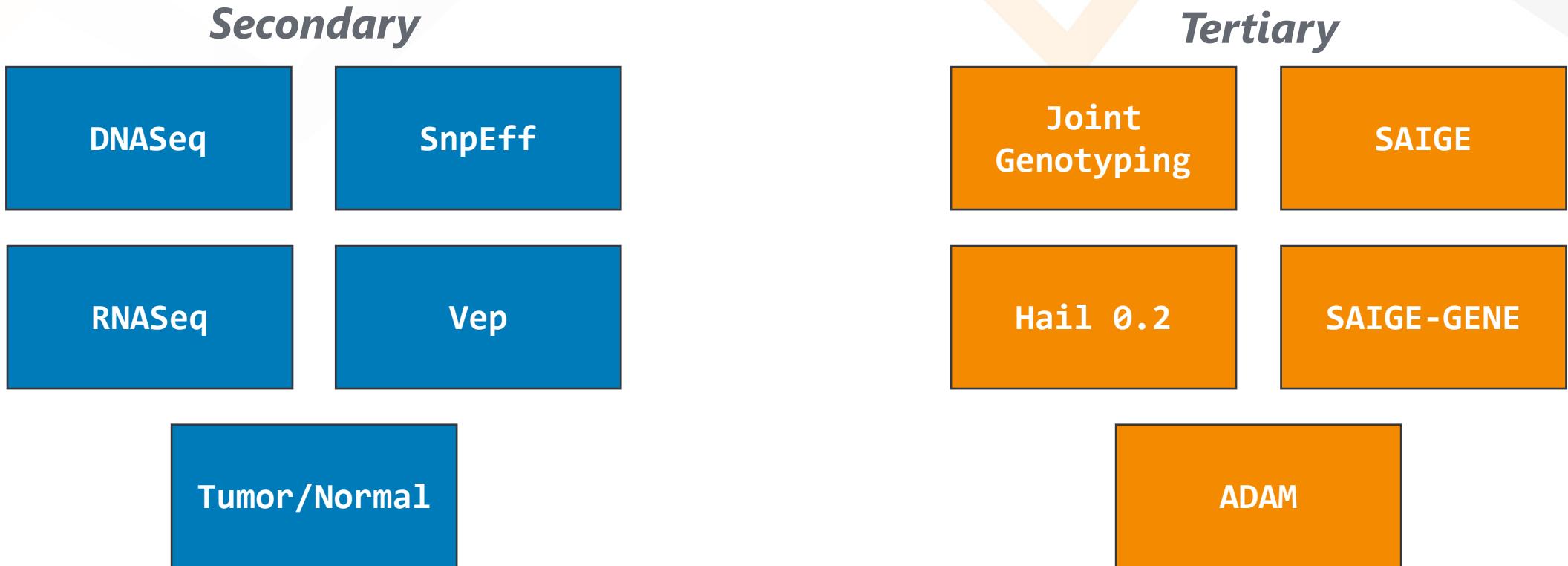
path	name	size
1 dbfs:/tmp/attributeanalysis/	attributeanalysis/	0
2 dbfs:/tmp/ccdaikest/	ccdaikest/	0
3 dbfs:/tmp/dataframe_sample.csv	dataframe_sample.csv	272
4 dbfs:/tmp/dbamhart@blue-granite.com/	dbamhart@blue-granite.com/	0
5 dbfs:/tmp/genomics/	genomics/	0
6 dbfs:/tmp/hive/	hive/	0
7 dbfs:/tmp/jcarlson@blue-granite.com/	jcarlson@blue-granite.com/	0

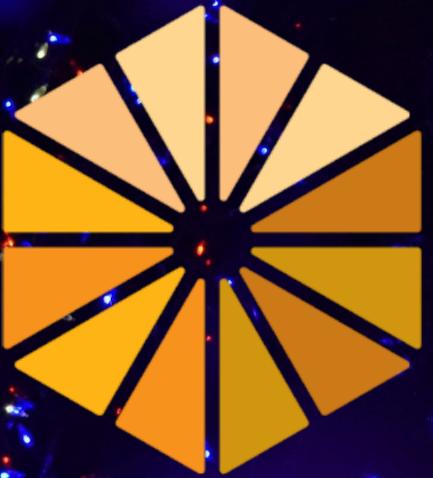
'Cmd 3' contains Scala code for importing a pipeline class:

```
1 import com.databricks.hls.pipeline.dnaseq._  
2 import com.databricks.hls.pipeline._  
3 import com.databricks.hls.pipeline.dnaseq.annotation.AnnotationPipeline  
4  
5 val pipeline = AnnotationPipeline
```

- Familiar notebook-style IDE
- Massively scalable
- Collaborative
- Python, R, Scala, and SQL

# Genomics Runtime - Analysis Pipelines





**GLOW**



# GLOW

*Analysis Tools*

- Read and Write *VCF*, *Plink*, and *BGEN* Files
- Read Genome Annotations (*GFF3*)
- Perform Variant Quality Control
- Perform *liftOver* Genomic Conversions
- Perform Variant Normalization
- Split Multiallelic Variants
- Prepare Genomic Data for Machine Learning
- Parallelize Common Tools with a Transformer
- Utilize Python Statistics Libraries
- Perform GWAS Regression Tests



# Azure Machine Learning Service

Enterprise-grade machine learning service to build and deploy models faster.

The screenshot shows the Microsoft Azure Machine Learning studio interface. The left sidebar contains navigation links for Home, Notebooks, Automated ML (preview), Designer (preview), Assets, Datasets, Experiments, Pipelines, Models, and Endpoints. The main area features a "Welcome to the studio!" section with four buttons: "Create new" (Notebooks, Automated ML (preview), Designer (preview)), "Notebooks" (with "Start now" button), "Automated ML (preview)" (with "Start now" button), and "Designer (preview)" (with "Start now" button). Below this is a "My recent resources" section showing a "Compute" instance named "cford-1-ci" (Type: Compute instance, Provisioning state: Succeeded, Created on: May 14, 2020 3:23 PM). A "View all compute →" link is also present. At the bottom, there is a "Tutorials" section with six items: "What is Azure Machine Learning?", "Train your first ML model with Notebook", "Create, explore and deploy Automated ML experiments.", "What is Azure Machine Learning designer?", "What are compute targets in Azure Machine Learning?", and "Deploy models with Azure Machine Learning".

- Familiar JupyterLab, Jupyter, and RStudio IDEs
- Python and R SDKs
- Easy operationalization of code as APIs



# Azure Genomics Service

Power genome sequencing and research insights.

Home > New > Marketplace > Everything > Genomics > Create

## Create a Genomics account

[Basics](#) [Tags](#) [Review + Create](#)

Microsoft Genomics service provides a cloud hosted solution that makes it easy to analyze genomic samples. The service takes in genomic samples as two paired end read fastq (.fq.gz) files and associated log files. The processing uses a BWA / GATK best practices pipeline to produce results faster and with greater accuracy. By using the Microsoft Genomics service, you can accomplish simply, consistently, and reliably, you are able to focus on your research.

[Learn more](#)

**PROJECT DETAILS**

Select the subscription to manage deployed resources and costs. Use the dropdown menu to choose a resource group or create a new one to manage all your resources.

\* Subscription:   
└─ \* Resource group:  [Create new](#)

**INSTANCE DETAILS**

\* Account name:  Enter the name  
\* Location:

# Microsoft Genomics service - Command Line Interface - Configuration File  
# Documentation: <https://docs.microsoft.com/azure/genomics/>  
# Instructions  
# 1. Entries are provided in key - value pairs, like key: value  
# 2. Whitespace (tabs, spaces) don't matter  
# 3. Lines starting with # are ignored  
  
# Example usage:  
#  
# pip install msgen  
# msgen submit -f c:\temp\config.txt -b1 sample\_1.fq.gz -b2 sample\_2.fq.gz  
# msgen submit -f c:\temp\config.txt -b1 sample.bam  
  
api\_url\_base: <Your Genomics Service API url here>  
access\_key: <Your Genomics account key here>  
  
process\_args: # Other available references (replace hg19m1 below): b37m1, hg19m1, hg38m1, hg38m1x  
R=hg19m1  
  
# Uncomment the appropriate process\_name  
process\_name: snappatk  
#process\_name: gatk4  
  
poll: false  
  
# To learn more about the optional "emit\_ref\_confidence" argument, see <https://github.com/microsoft/msgen#release-notes-v080>  
# Uncomment the "emit\_ref\_confidence" argument below to produce "g.vcf" outputs.  
#emit\_ref\_confidence: gvcf  
  
# To learn more about the optional "bgzip\_output" argument, see <https://github.com/microsoft/msgen#release-notes-v090>  
# Uncomment the "bgzip\_output" argument below to produce "\*.vcf.gz" and "\*.vcf.gz.tbi" outputs.  
#bgzip\_output: true  
  
input\_storage\_account\_name:  
input\_storage\_account\_key:  
input\_storage\_account\_container:  
output\_storage\_account\_name:  
output\_storage\_account\_key:  
output\_storage\_account\_container:

[Review + Create](#) [Next: Tags](#)

- Python 2-based API
- Cloud implementation of Burrows-Wheeler Aligner (BWA) and the Genome Analysis Toolkit (GATK) for secondary analysis
- Uses FASTQ or BAM inputs

⚠ This service is a bit outdated and unmaintained by Microsoft at this point

*But what  
about storing  
and organizing  
data in its  
many forms?*



*Azure Storage  
(Data Lake)*

Petabyte-size file storage with RBAC, auditing, and massive scalability

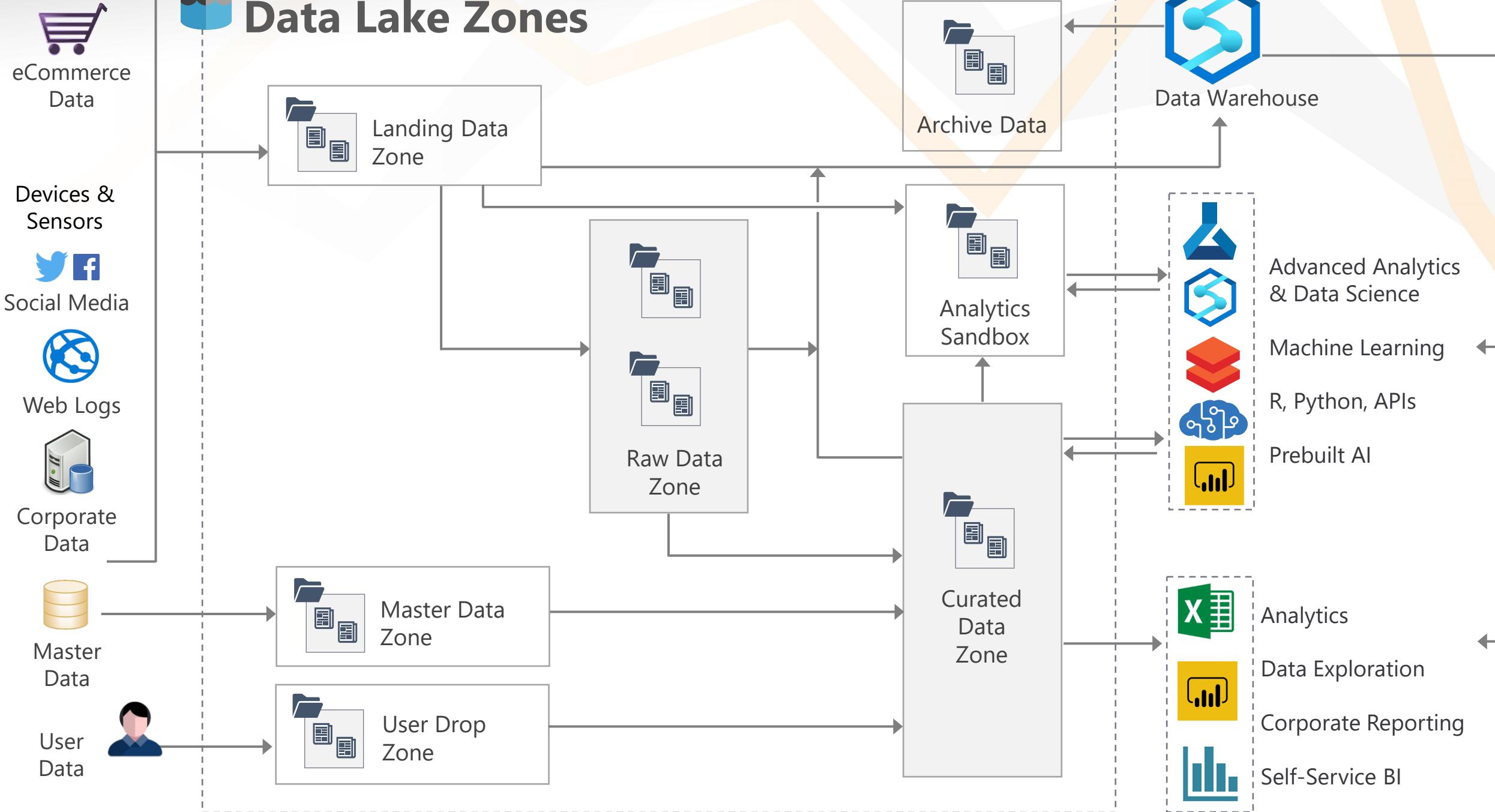
+



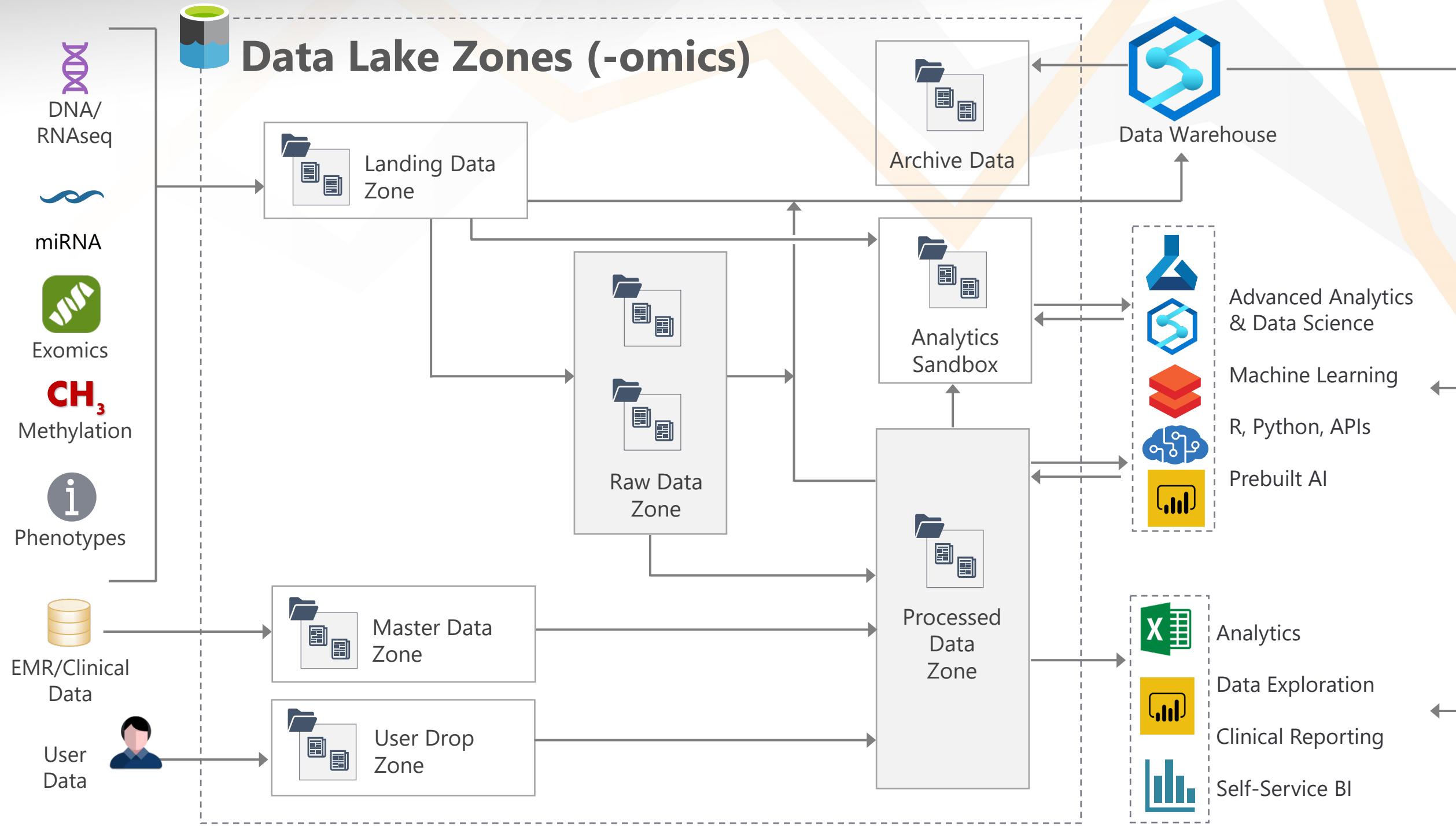
*Delta Lake*

Open-source storage layer that brings ACID transactions to big data workloads.

# Data Lake Zones

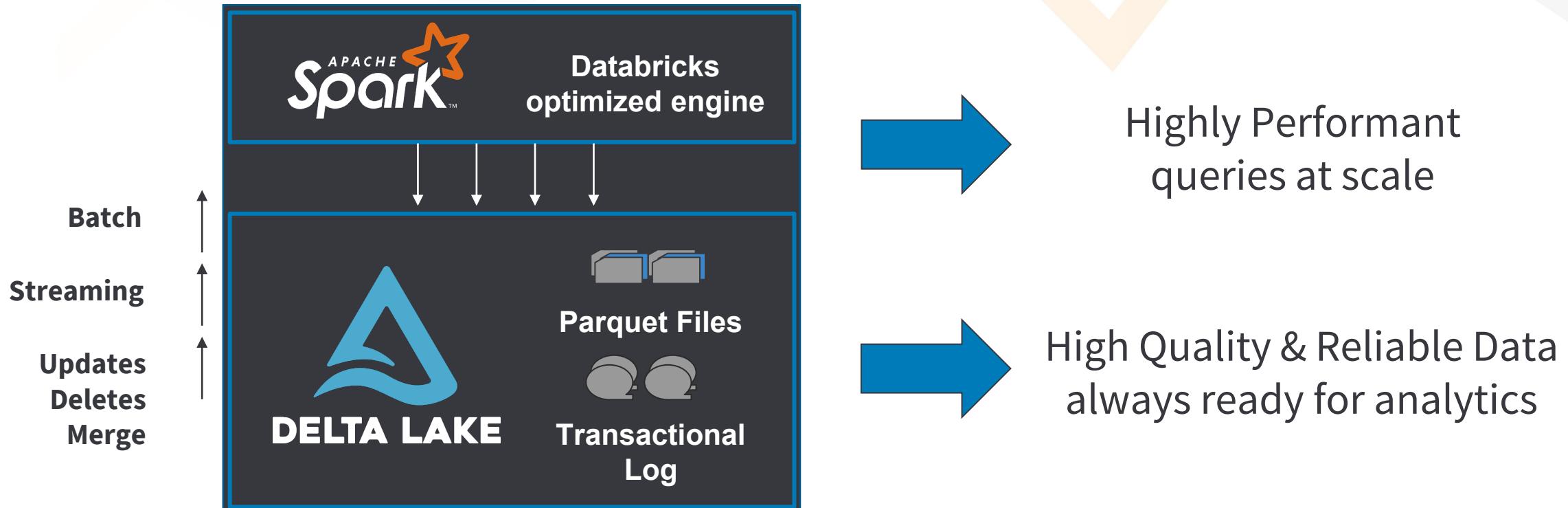


# Data Lake Zones (-omics)



# Delta Lake Format

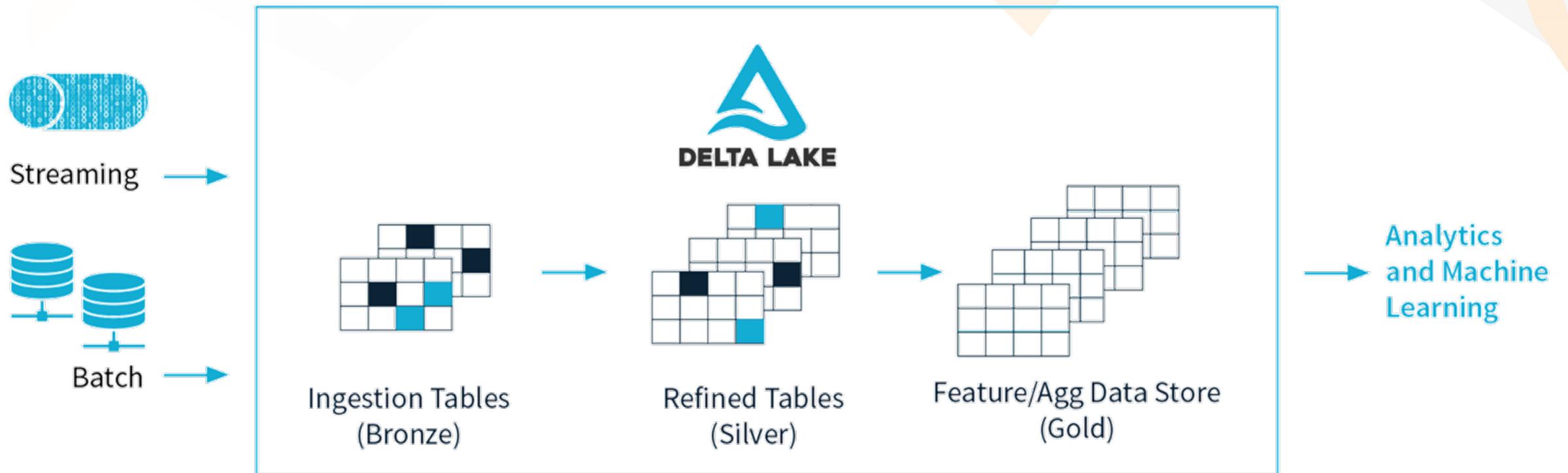
New Open Source Format Based on Parquet



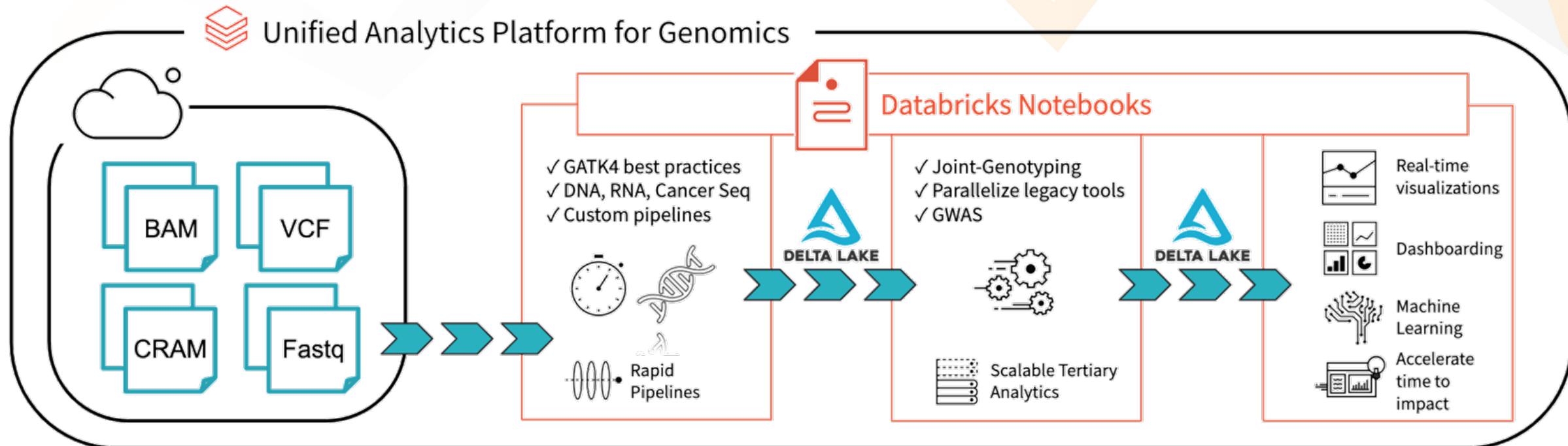
## Key Features

- ACID Transactions
- Schema Enforcement
- Unified Batch & Streaming
- Time Travel/Data Snapshots

- Indexing
- Compaction
- Data skipping
- Caching



# Delta Lake Workflow and Architecture



# Delta Lake Workflow and Architecture

