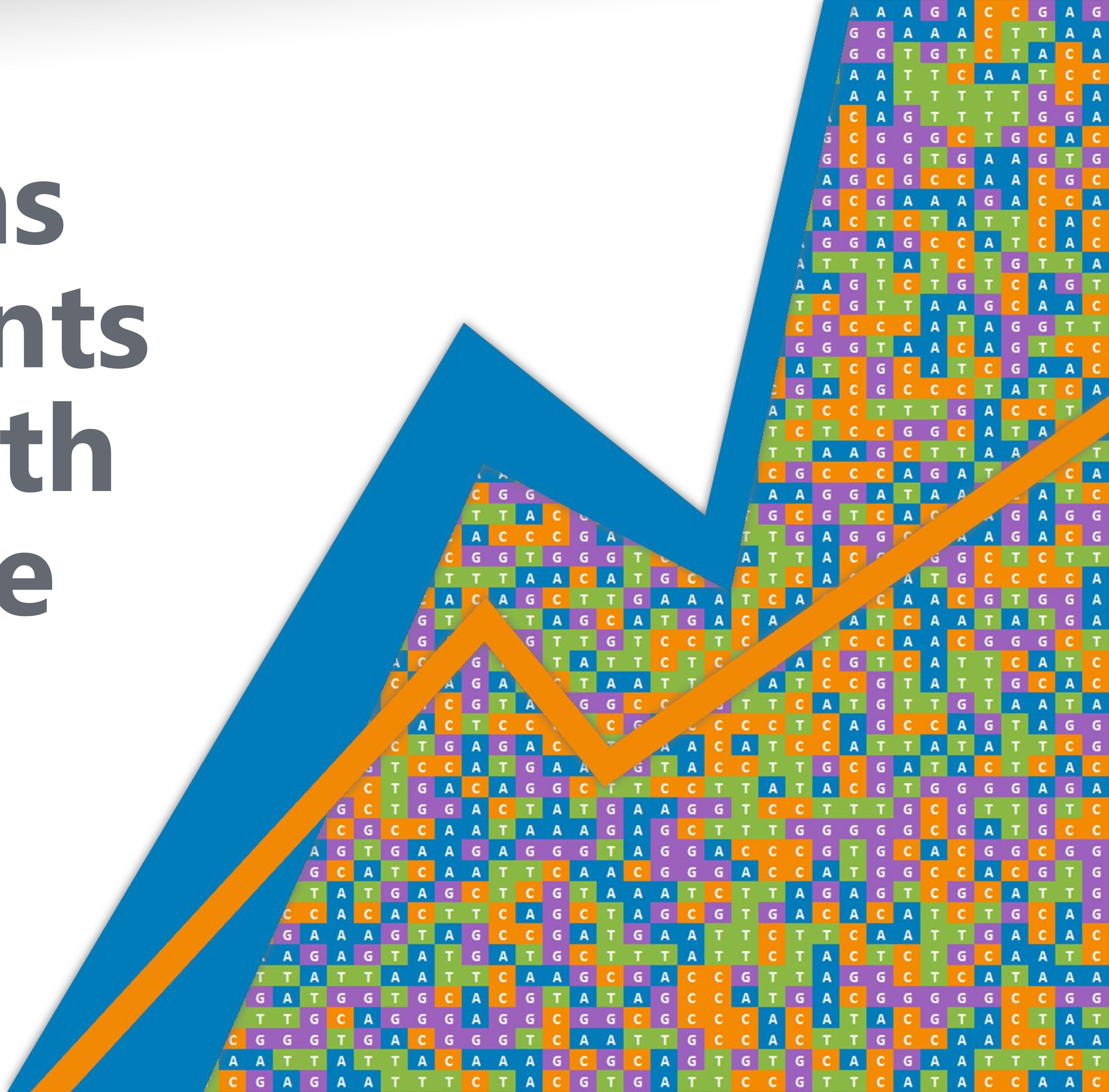
Query Millions of WGS Variants in Minutes with Azure Synapse

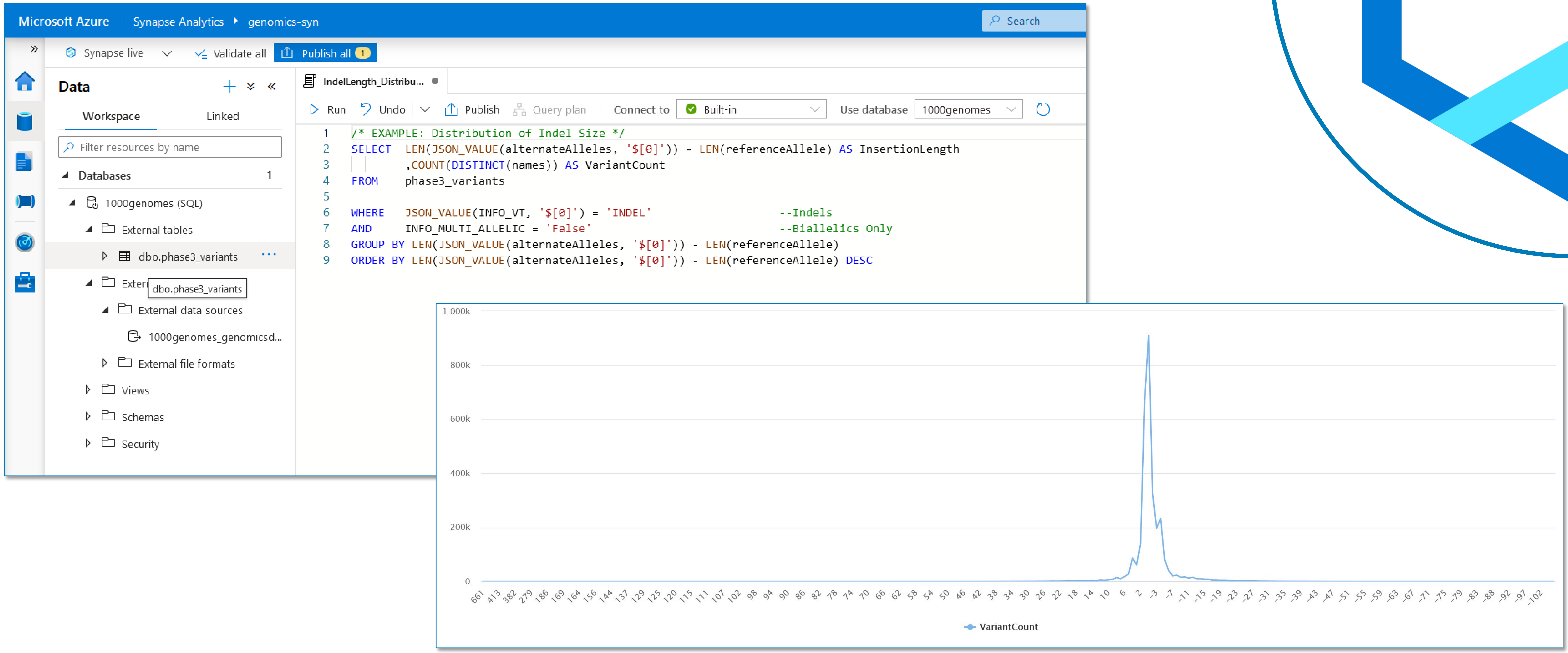


BLUEGRANITE
DATA • INSIGHTS • ANALYTICS

What’s Azure Synapse?

“Limitless analytics service with unmatched time to insight.”

- Scalable analytics platform that includes a SQL engine (formerly Azure SQL Data Warehouse), Data Lake exploration, Apache Spark, integration with Power BI, and more.



```sql
/* EXAMPLE: Distribution of Indel Size */
SELECT  LEN(JSON_VALUE(alternateAlleles, '$[0]')) - LEN(referenceAllele) AS InsertionLength
        ,COUNT(DISTINCT(names)) AS VariantCount
FROM    phase3_variants

WHERE   JSON_VALUE(INFO_VT, '$[0]') = 'INDEL'                   --Indels
AND     INFO_MULTI_ALLELIC = 'False'                            --Biallelics Only
GROUP BY LEN(JSON_VALUE(alternateAlleles, '$[0]')) - LEN(referenceAllele)
ORDER BY LEN(JSON_VALUE(alternateAlleles, '$[0]')) - LEN(referenceAllele) DESC
```

Quick Data Stats

1000 Genomes Project

80+ Million Variants

2,504 Individuals

VCF File Size: ~168GB

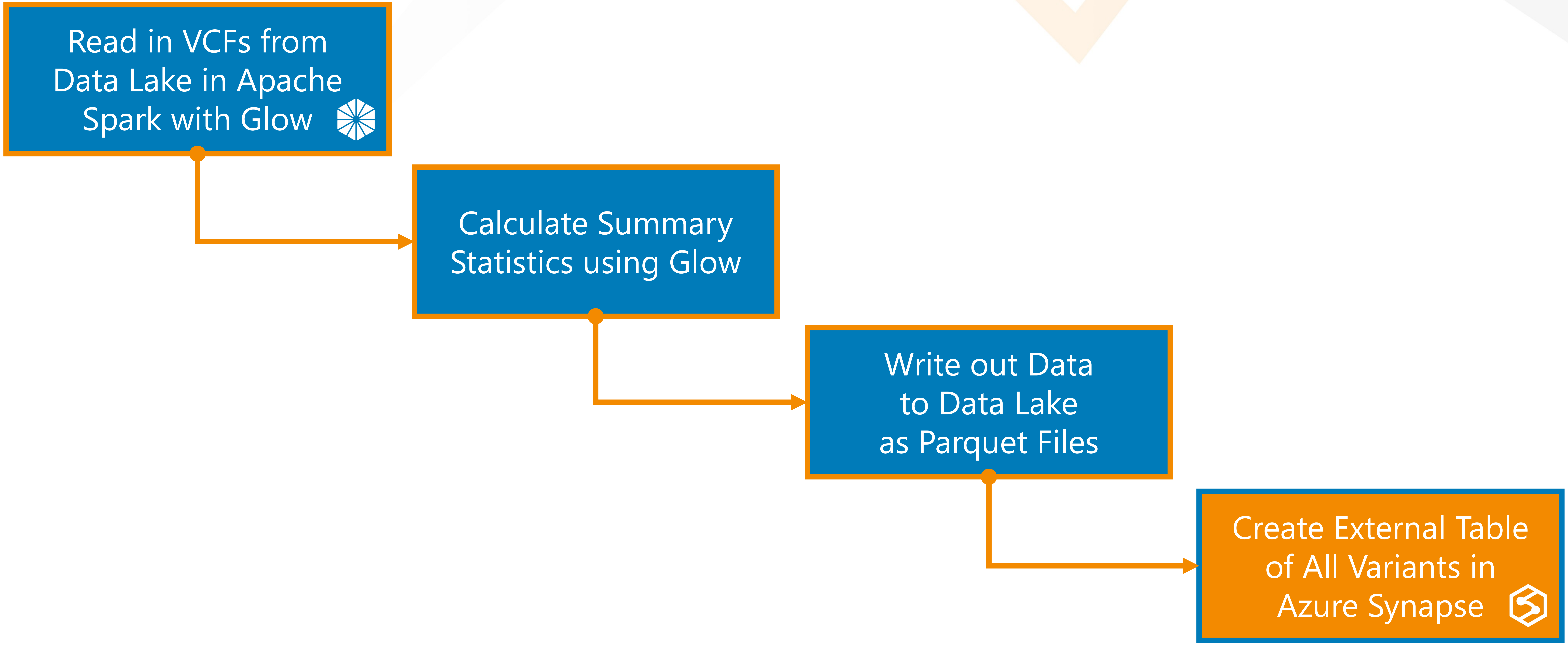
Parquet File Size: ~75GB

Phase 3 (GRCh38)

Population
Yoruba
Luhya
Dai Chinese
CEPH
Japanese
Han Chinese
Gujarati
Tamil
Telugu
British
African Caribbean
Puerto Rican
Southern Han Chinese
Finnish
Kinh Vietnamese
Bengali
African Ancestry SW
Colombian
Peruvian
Punjabi
Mende
Esan
Gambian Mandinka
Iberian
Toscani
Mexican Ancestry

Sudmant, P., Rausch, T., Gardner, E. *et al.* An integrated map of structural variation in 2,504 human genomes. *Nature* **526,** 75–81 (2015). https://doi.org/10.1038/nature15394

Pipeline



Read in VCFs from Data Lake in Apache Spark with Glow

Calculate Summary Statistics using Glow

Write out Data to Data Lake as Parquet Files

Create External Table of All Variants in Azure Synapse

Converting VCFs to Parquet Files

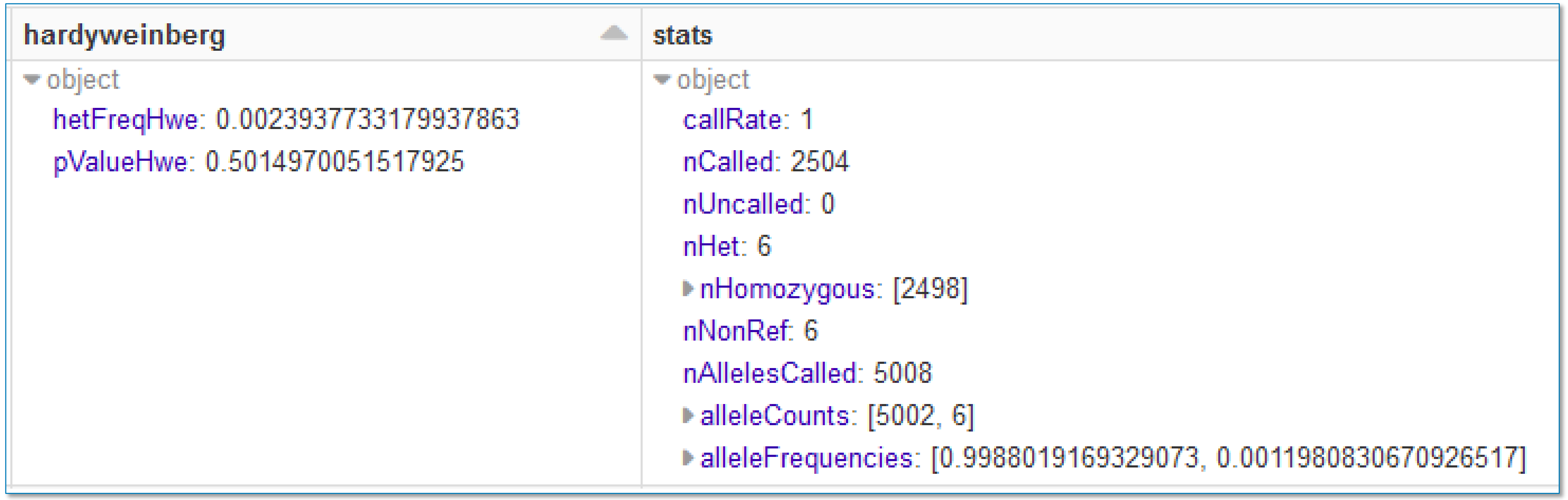
(in 4 lines of code)

```
input_vcf_path = "/mnt/1000genomes/phase3_vcfs/chr1.vcf.gz"
output_parquet_path = "/mnt/1000genomes/phase3_parquets/chr1.parquet"

vcf_df = spark.read.format("vcf").load(input_vcf_path) \
               .withColumn("hardyweinberg", expr("hardy_weinberg(genotypes)")) \
               .withColumn("stats", expr("call_summary_stats(genotypes)"))

vcf_df.write.format("parquet").save(output_parquet_path)
```

+ 2 lines for adding some stats…



```
hardyweinberg                              stats
▾ object                                   ▾ object
   hetFreqHwe: 0.0023937733179937863          callRate: 1
   pValueHwe: 0.5014970051517925              nCalled: 2504
                                              nUncalled: 0
                                              nHet: 6
                                              ▸ nHomozygous: [2498]
                                              nNonRef: 6
                                              nAllelesCalled: 5008
                                              ▸ alleleCounts: [5002, 6]
                                              ▸ alleleFrequencies: [0.9988019169329073, 0.0011980830670926517]
```