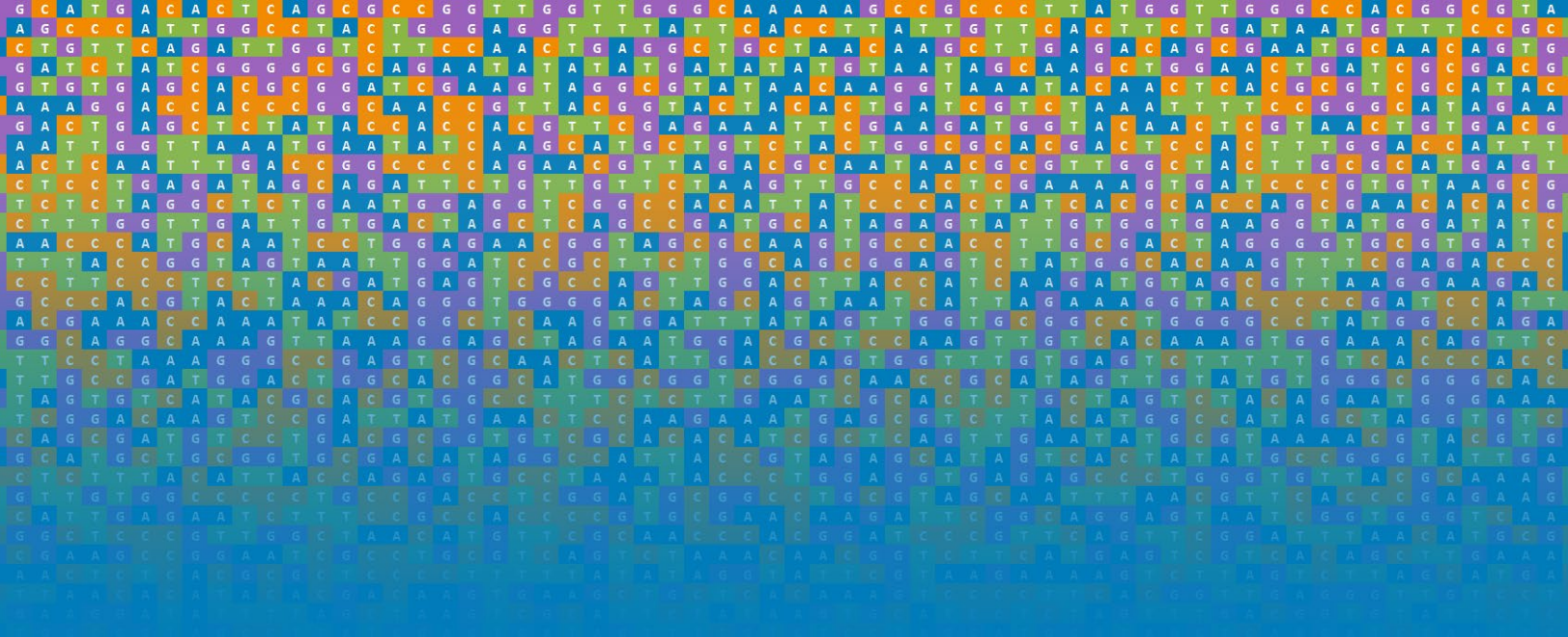


Building a Genomics Data Lake in Azure

Scalability • Security • Collaboration

By: Colby T. Ford, Ph.D.
and Larry Baker

BLUEGRANITE

"IN THE LONG HISTORY OF HUMANKIND (AND ANIMAL KIND, TOO) THOSE WHO LEARNED TO COLLABORATE AND IMPROVISE MOST EFFECTIVELY HAVE PREVAILED."

- CHARLES DARWIN

"EACH OF US HAVE THINGS AND THOUGHTS AND DESCRIPTIONS OF AN AMAZING UNIVERSE IN OUR POSSESSION THAT KINGS IN THE 17TH CENTURY WOULD HAVE GONE TO WAR TO POSSESS."

- KARY MULLIS

BLUEGRANITE.COM/GENOMICS

First publication, January 2021

Contents

Introduction

It's hard to believe that the first draft of the human genome was completed 19 years ago[1]. In 2001, we had high hopes for what this body of knowledge would unlock in the understanding of our own being. Nearly two decades later, researchers like you are still innovating, creating new ways to sequence and analyze genomic data in quantities and velocities unlike anything we have previously experienced. As a result, we now see frequent and astonishing advancements in precision medicine, understanding diseases, developing treatments and vaccines, and more.

It is now estimated that, by 2025, we will use 40 exabytes of storage to house human genomic data[2]. To put that growth into perspective, 5 exabytes can hold all the words ever spoken by humans[3]. In addition, we can assume that compute needs to process this vast amount of data will likewise continue to grow. This estimate does not even begin to factor genomic data from other organisms, but we can expect the growth of genomics research in plant biology, infectious diseases, and other species will certainly follow suit.

Given this flood of data, research organizations have two options: 1) continue to purchase large amounts of computing and storage equipment, or 2) utilize the flexibility and scalability of the cloud to do the heavy lifting.

With Microsoft Azure, organizations across industries have been experiencing the power of harnessing the cloud to do more with their data. The cloud offers a secure and flexible environment to easily spin up storage or compute services without having to manage the underlying architecture, reducing the time to gain insights from your data. The genomics field, however, is a little behind the curve when it comes to taking advantage of cloud computing services.

In this book, we will discuss the utility of Azure Data Lake and how this flexible storage option promotes collaboration and scalability in your genomics practice while also ensuring a secure and stable environment for your genomics data. Plus, with its easy integration with other Azure services, orchestrating and automating data movement and bioinformatics pipelines has never been easier (or faster).

About Azure Data Lake

Azure Data Lake Storage (ADLS) is a massively scalable and secure data storage option that is perfect for housing genomics data. When you begin interacting with it, it will seem familiar, almost like Microsoft Office OneDrive, Google Drive, or Dropbox. You can store practically all types of files of any size in ADLS and organize them how you like.

At its core, ADLS is built on top of Azure Blob Storage, a popular storage option for unstructured and big data solutions, but with some exciting features on top.

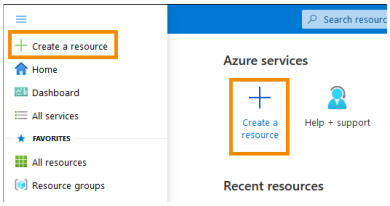
ADLS provides a *hierarchical namespace* which allows us to organize files in a familiar manner in the cloud. More specifically, this allows us to understand our data lake's organization as if it is any other file system that we are used to on our local desktop computer.

ADLS also is optimized for performance. When you are ready to begin analyzing your data in Azure, simply point your compute service to the data lake to serve up your data efficiently. This performance gain is exemplified when it comes to distributed services that use the Hadoop FileSystem, such as Azure Databricks, Azure Synapse Analytics, or Azure HDInsight.

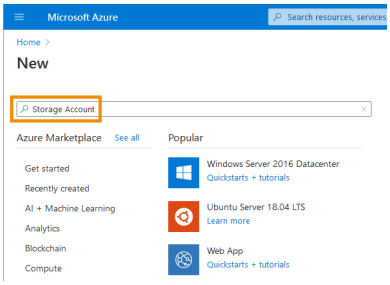
To learn more about Azure Data Lake Storage, visit https://azure.microsoft.com/en-us/services/storage/data-lake-storage/.

Setting Up a Data Lake in Azure

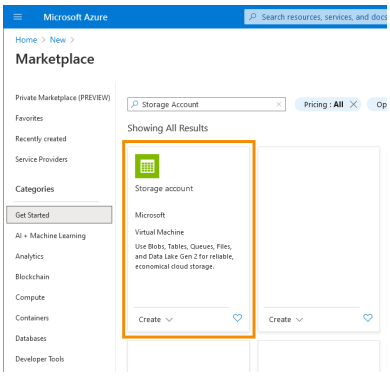
To begin setting up your genomics data lake, let's start by provisioning your Azure Data Lake service. Assuming you already have an Azure tenant and subscription, navigate to the Azure Portal. Once there, click on **+ Create a resource** either at the top of the screen or from the menu on the left.



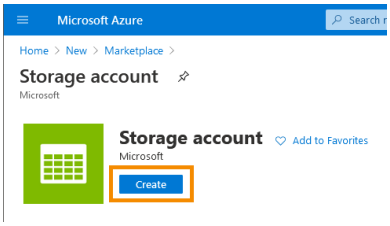
From the **New** screen, search for "Storage Account". Select the **Storage Account** option from Microsoft.



Click **Create** on the next screen.

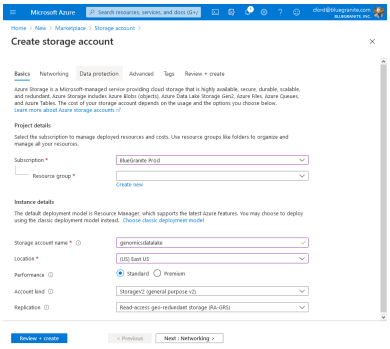


On the **Basics** tab in the form, select your desired Subscription, select (or create) your desired resource group, and pick a Storage account name.



As for the rest of the options, here are some recommendations for the next few tabs:

- Basics
  - Location: Pick the region that is closest to you or the region closest to your other large data sources.
  - Performance: This option is a cost vs. speed question. Standard accounts use traditional (slower) magnetic drives, but are cheaper and better for bulk amounts of infrequently accessed storage. Premium accounts use solid state drives and are faster and provide lower latency.
  - Account kind: Use "StorageV2".
  - Replication: This option dictates where your data is replicated outside of its primary location. To learn more about replication options, visit azure.microsoft.com/documentation/articles/storage-redundancy/

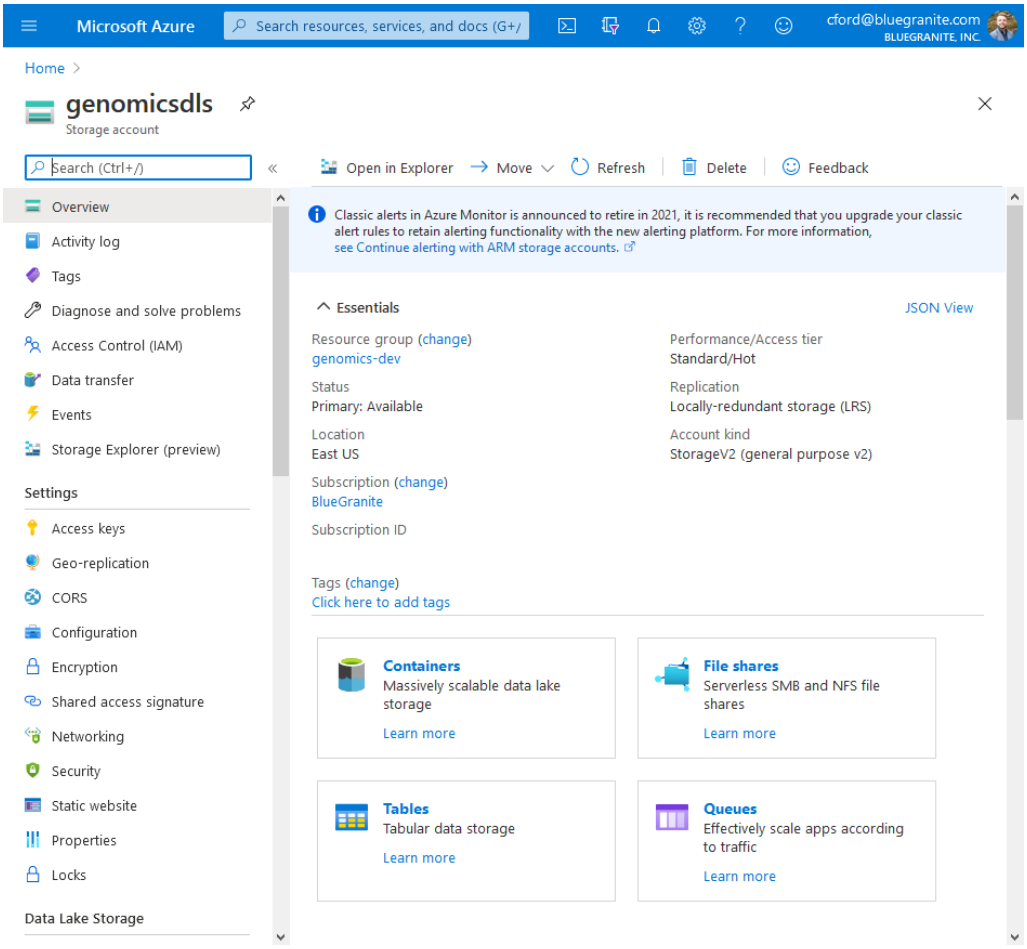


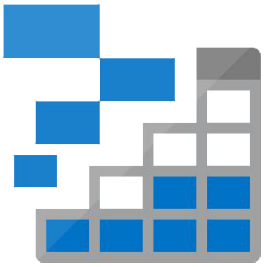
- Networking
  - Connectivity method: Depending on your security requirements, you may need to only access the data lake from a private endpoint. If you are unsure, select "Public endpoint (selected networks)" for now.
  - Routing preferences: Use "Microsoft network routing".
- Data protection
  (If you plan on housing multiple terabytes of data in your data lake, leave these options off. Since using a hierarchical namespace is recommended, you cannot use these features.)

  - Recovery and Tracking: Point-in-time restore, soft delete, and versioning are great features for rolling back changes or deletions in your data lake or retaining multiple historical versions of your data.

- Advanced
  - Security: Use default settings.
  - Blob storage: Use default settings.
  - Data Lake Storage Gen2: Enable Hierarchical namespace.

Once you have selected your desired options for your data lake, navigate to the last tab called **Review + create**. Once the Portal performs some validation checks on your selections, you can then click the **Create** button at the bottom. The data lake should only take a few minutes to provision.





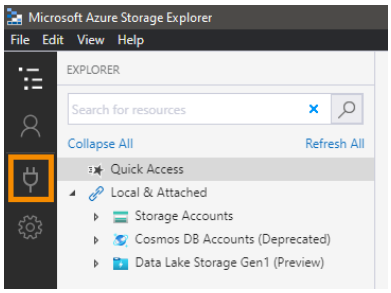
Azure Storage Explorer

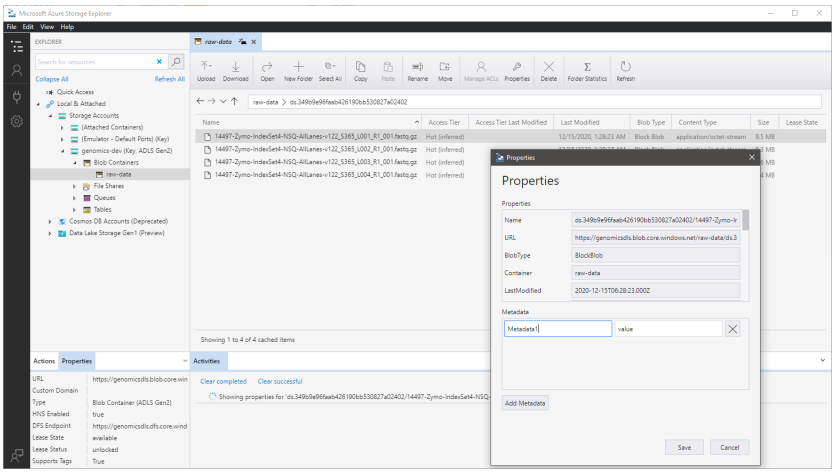
Azure Storage Explorer provides an easy-to-use graphical interface for interacting with your data lake. This tool, provided by Microsoft, allows users to browse, search, open, download, and upload files into their data lake in Azure. This tool is also useful in setting access control policies on individual files or folders.

To begin, download Azure Storage Explorer from azure.microsoft.com/features/storage-explorer/.

Once installed, click on the plug icon on the left sidebar and follow the prompts to connect.

After you have logged in, you can now peruse your data lake with ease.





AzCopy

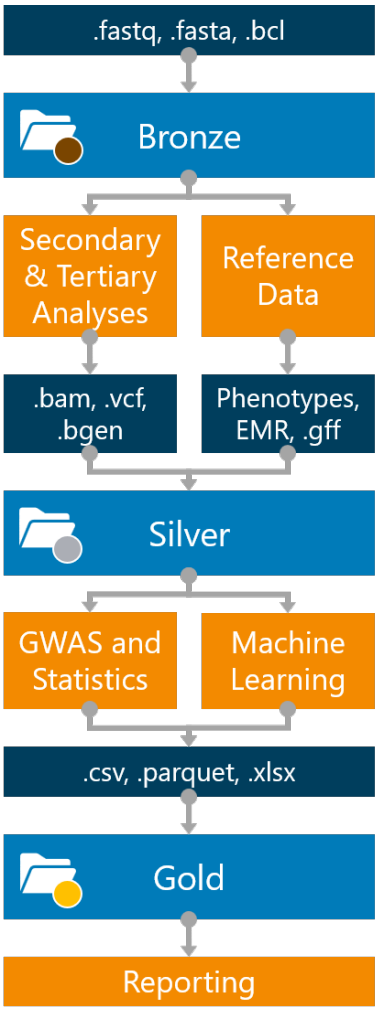
AzCopy is a command-line tool for copying data to and from Azure Data Lake. This simple tool makes it easy to copy files between your local file system and your cloud data lake and even works well on rather large files. This is useful for scenarios where manual movement of data to and from the cloud is unavoidable. To begin, download AzCopy from docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10.

Once installed, using the tool is very similar to the `cp` command in Linux. First, login to your Azure account from the command line using `azcopy login` and then run: `azcopy copy <local-file-path> <data-lake-uri>` to copy a file up to the data lake.

To download a file, simply switch the order of the file path and the data lake URI: `azcopy copy <data-lake-uri> <local-file-path>`.

Organizing for Genomics

In a data lake, organization is key as it sets the standard for placing your data in the appropriate locations such that it is easily retrievable, cataloged, and can be secured appropriately. In reference architectures, you may see data lakes separated by zones dictating which data goes together, its purpose, and who should be able to access it. Another way to think about this is to categorize your data by its level of preparation into Bronze, Silver, and Gold zones.



Bronze

The Bronze area of a data lake serves as the ingestion point for raw data, which is often the unprocessed data from source systems. In genomics, this often translates to the place where you keep the data that comes directly from your sequencing process. This could be multiplexed sample files or outputs from an Illumina sequencing run (such as `.bcl` files).

Depending on your genomics practice, your raw data may instead be individual samples (`.fasta` or `.fastq` files) or accompanying data. Either way, this zone serves as the starting point to contain your data in its most basic form.

Silver

Continuing on, the Silver area of the data lake serves multiple purposes. It houses data that is a bit more refined, curated, or enriched. For genomics, this zone should house your data that has been processed such as data resulting from secondary and tertiary analyses. This includes alignments (`.sam` or `.bam` files), the results of variant analyses (`.vcf`), annotations (`.gff`), population analyses (`.bgen`), or results from analyses seen in other *-omics* areas (e.g., phylogenetic trees, `.allc`, or `.pdb` files).

In addition, this zone can also include accessory data that is used to enrich your other data. For example, this could be phenotype information or maybe electronic medical records (EMR), any data that can be used to augment the insights from the main genetic information. The purpose of the Silver zone is to serve analytical use cases such as downstream machine learning or statistical analyses.

Gold

Finally, the Gold zone houses data in its most cleansed and aggregate form. This is often the resulting data from other workflows, outputs from machine learning or statistical tasks, genome-wide association studies (GWAS), and research findings. Data types in this zone may be in less bioinformatics-specific formats (such as `.csv` files, Excel workbooks, or plots) or formats to store larger files types (such as `.parquet`, `.avro`, or `.delta`).

Traditionally, the purpose of the Gold zone is to serve reporting needs and business-critical use cases. In genomics, it should showcase results of any prior analyses and be the final source of truth for any workflows.

Service Tiers

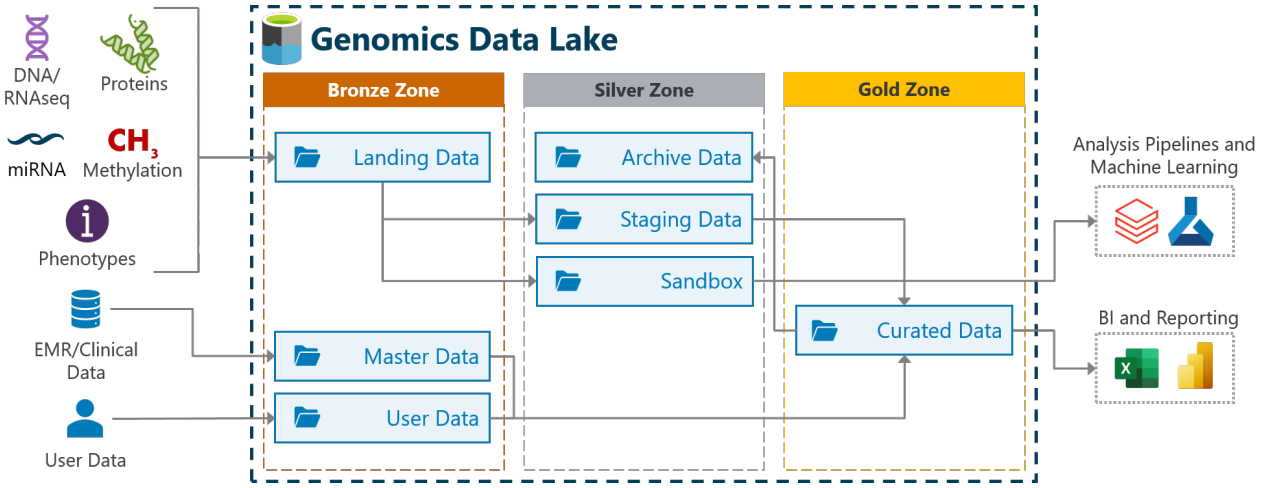
In Azure Data Lake, there are three access tiers that provide varying levels of accessibility for your data at different price points. This is useful in genomics as file sizes start to grow, certain pieces of data may be useful to keep, but might not be often used. As such, some consideration should be given around migrating large amounts of unused data to a lower storage tier to save on cost.

If, when setting up your data lake, you selected the Standard performance tier, these are the following access tier options:

- Hot - Use for data that is accessed frequently. This is the best tier for most data scenarios.
- Cool - Use for data that is infrequently accessed (less than monthly).
- Archive - Only use for data that is very infrequently accessed (less than yearly). This tier should only be used on archived data or backups.

If, when setting up your data lake, you selected the Premium performance tier, your access tier will always be optimized for low latency. This is ideal for interactive workloads surrounding large-scale analytical, data transformation, or machine learning tasks.

While cost is certainly a driving factor for selecting a service tier, it should be noted that getting data out of lower service tier may incur its own transaction cost and may be subject to a delay before that data is available for use.



Delta Lake

Recently, the developers at Databricks released a new set of capabilities to overlay on top of a data lake. This new functionality, called Delta Lake[4], provides ACID transactions to parquet files in a data lake. This means that you can now get transaction attributes (atomicity, consistency, isolation, durability) to help guarantee data validity and make your data platform more resilient against error. Prior to Delta Lake, this transaction functionality was only commonly seen in relational databases.

In addition, Delta Lake provides a scalable way to manage metadata, an open-source format (`.delta`), and is 100% compatible with the Apache Spark API, making it a perfect choice for scalable bioinformatics use cases.

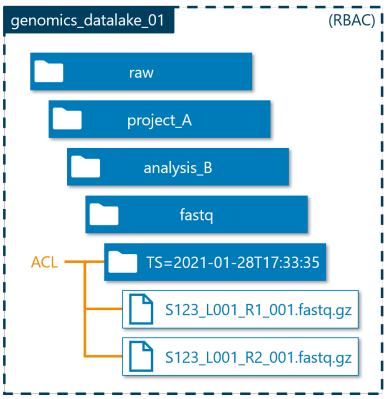
One feature that may be important for research organizations is that Delta Lake provides "time travel" or data versioning so that users can revert changes. This allows for improved auditing and better reproducibility for large-scale experiments. Plus, Delta Lake offers features for handling metadata, enforcing and updating schemas, and more.

To learn more about Delta Lake, visit delta.io.

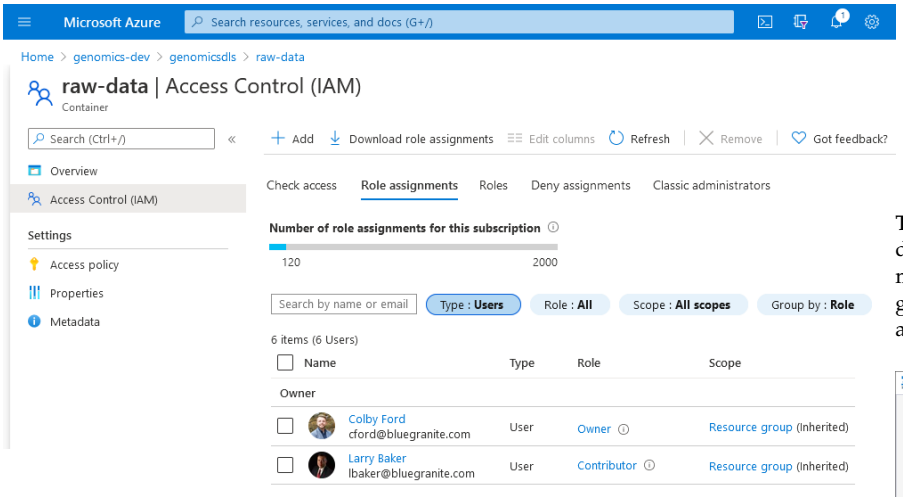
Security Considerations

One common selling point for using Azure Data Lake over the more basic Azure Blob Storage is that you get finer-grained controls over who can access data. This is set at two levels within an account: 1) at the container level with Role-Based Access Controls (RBACs), 2) on individual directories or files using Access Control Lists (ACLs). Both of these types of access permissions are applied using your organization's Azure Active Directory Credentials.

Role-Based Access Controls are designed to apply a security principle to an entire storage account or container. Roles are easily set from the Azure Portal by selecting your data lake, then clicking **Access Control (IAM)** from the left side panel.

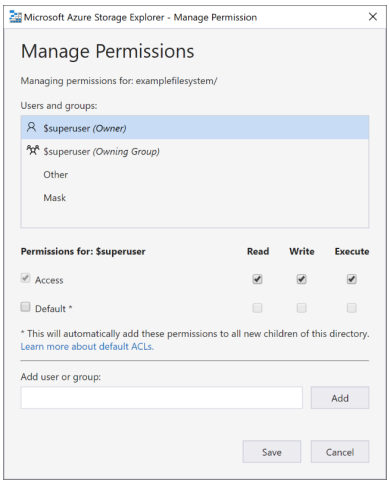


Storage Data Role	Owner	Contributor	Reader
Description	Gets full access to storage containers and data. Owners can modify the ACLs of all items.	Gets read, write, and delete access to storage containers and blobs. Contributors can modify the ACL of items.	Gets read and list access to storage containers and files.

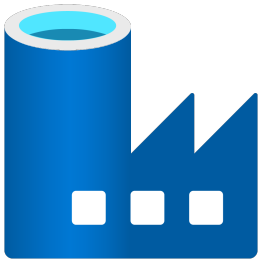


Tip: Set up groups of users under a defined role in Active Directory. It is much easier to manage access for a group than individuals, though both are supported.

Access Control Lists provide the ability to set a finer level of access to specific directories or files. ACLs can be set in a variety of ways, but the easiest is to use Azure Storage Explorer, select the file(s) or a directory that you want to manage, and click **Manage Access Control** Lists. You can then set read, write, and execute permissions for a single user or entire groups.



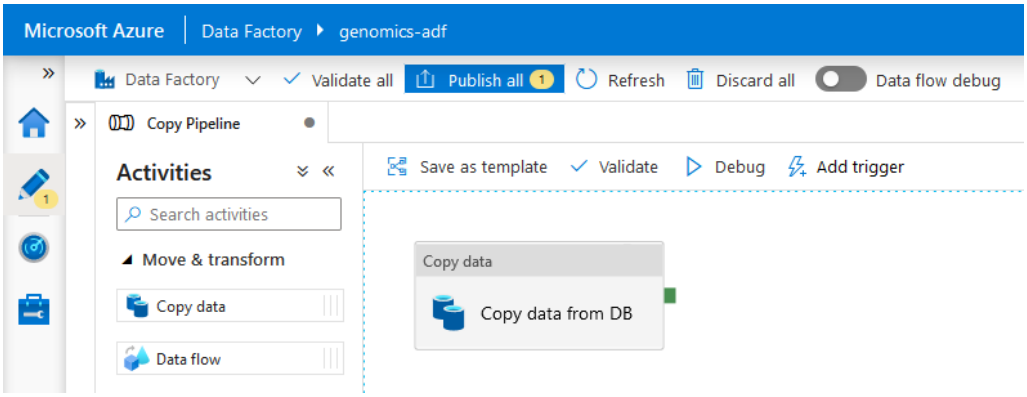
Data Orchestration



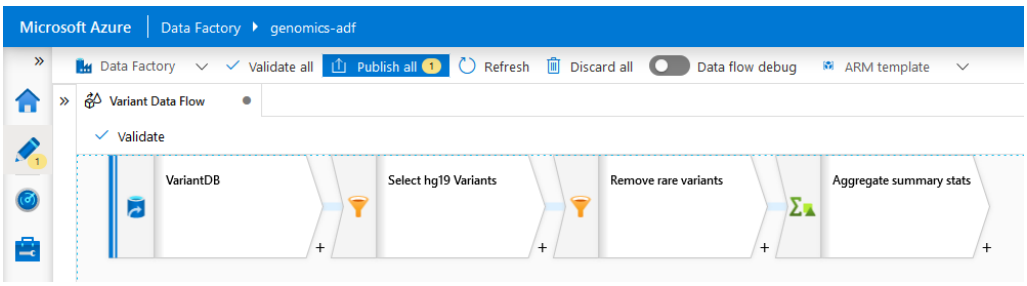
Automation with Azure Data Factory

Azure Data Factory is a fully managed data orchestration service that provides powerful options for moving data around, transforming it, and executing analyses on other services. Currently, Azure Data Factory includes over 90 data connectors to ingest your data from services like Microsoft SQL Server, Google BigQuery, Amazon S3, Teradata, FTP, REST APIs, and more.

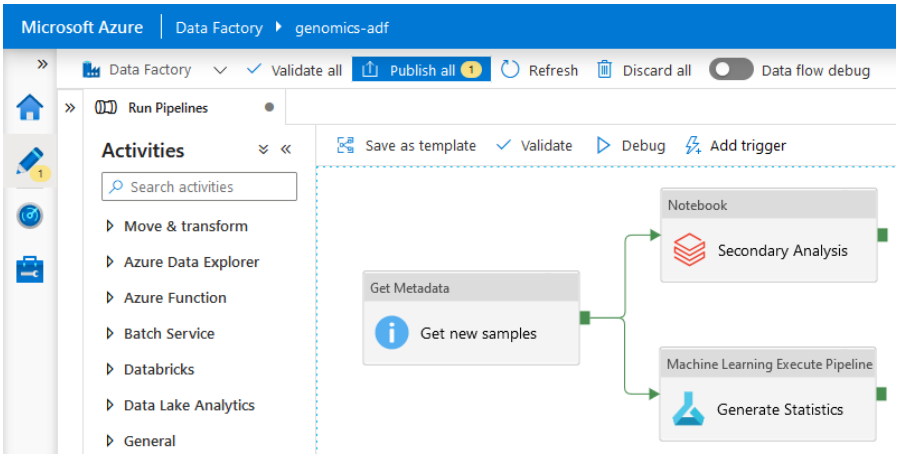
We often use Data Factory to copy data from source systems into Azure Data Lake. This is made easy with the Copy Data tool.



Or, for more advanced transformations, we can employ Data Flows to transform data as it is being moved from source to sink (destination).



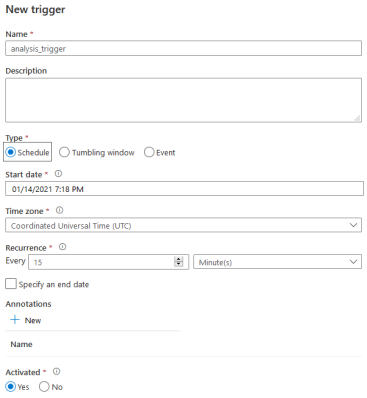
We can even call external services, such as Azure Databricks, Azure Machine Learning, and more, to analyze our data.



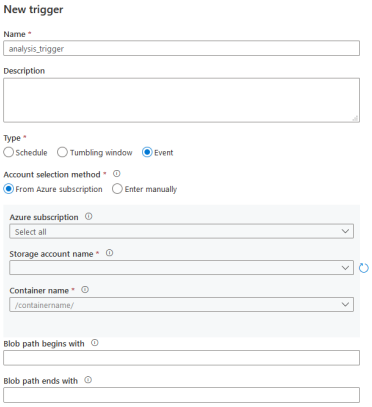
Plus, we can automate these pipelines such that little to no manual intervention is needed for continuous data pulls into your data lake. Pipelines can be triggered based on some schedule (for example, "Run the pipeline every 15 minutes") or from the occurrence of an event (for example, "When new data is available in the `samples` directory in the data lake, run this pipeline").

This automated approach for retrieving your data (such as project samples, analysis outputs, and other datasets) unlocks the ability to take advantage of the Azure cloud for secondary and tertiary analyses, machine learning, and more. For example, using Data Factory, we can execute secondary analysis pipelines in the Azure Databricks Runtime for Genomics, which allows for distributed jobs for alignment, filtering/quality control, variant calling, annotation, and more.

Scheduled:



Event-Triggered:



Illumina BaseSpace Connector

Today, many users use Illumina's online service called BaseSpace to house their sequencing data and to connect to third-party apps for downstream analyses. We at BlueGranite have taken this Data Factory functionality a step further in the genomics direction by creating a connector for Illumina BaseSpace.

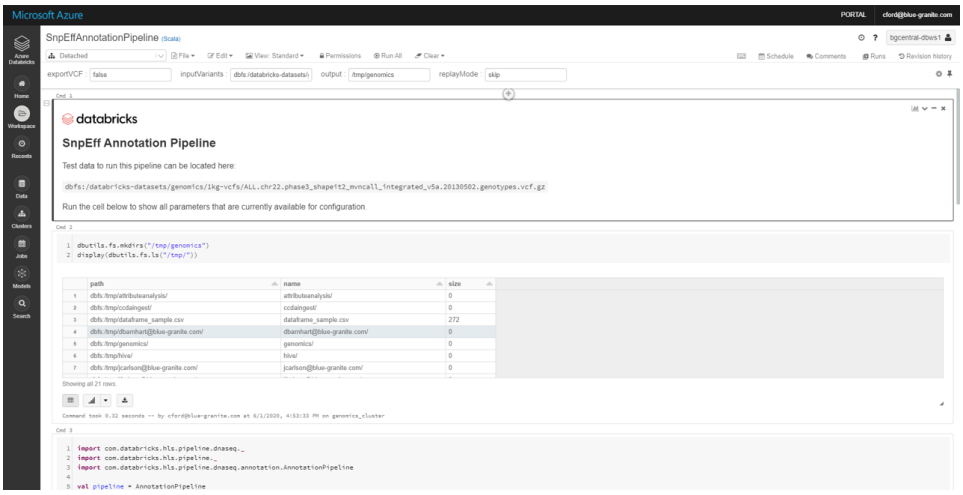
Like other connectors in Azure Data Factory, we can automate the ingestion and processing of data from various sources. With BaseSpace, we authenticate at the project level, so security around sensitive data stays in place. We then copy data from BaseSpace to Azure Data Lake, organizing the data by Project, Run, Dataset, and then data type.

Scaling Analyses

Azure Databricks

Azure Databricks is a managed Apache Spark[5] service for distributed computing. This service provides a fast, easy, and collaborative workspace for scaling big data analytics and machine learning workloads in the cloud. With Databricks, you can code in a language of your choosing (R, Python, SQL, or Scala) and use practically any open-source library along with the distributed Spark engine.

In addition to these standard data and machine learning features, Databricks also provides a specialized Runtime for Genomics, which provides an optimized compute environment for scaling bioinformatics analyses. The runtime provides pre-packaged, distributed pipelines for common secondary and tertiary analysis tasks, all optimized for scalability.



Pre-built Pipelines:

- Secondary Analysis
  - DNASeq pipeline
  - RNASeq pipeline
  - Tumor/Normal pipeline
  - Variant Annotation pipeline with SnpEff[6] or VEP[7]
- Tertiary Analysis
  - Joint Genotyping pipeline
  - Integration with Hail 0.2[8]
  - Pre-installed open-source tools such as ADAM[9], GATK[10], and SAIGE[11]

Glow

Also included with the Azure Databricks Runtime for Genomics is an open-source package known as Glow. Glow is a Spark-enabled package that is designed to help scale genomic workloads[12]. It was created by the genomics team at Databricks and the Regeneron Genetics Center. This package is enabled by default in the Runtime for Genomics, but can be used with other Apache Spark clusters as well.

Some notable functionality:

- Read and write common bioinformatics data types like VCF, PLINK, BGEN, and GFF3. (Similar to `spark.read.format("csv")`, read a VCF with `spark.read.format("vcf")`.)
- Handle variant-related tasks like quality control, normalization, and splitting or merging.
- GloWGR: Whole Genome Regression (a distributed version of regenie[13]).
- Easily parallelize some existing libraries using the Pipe Transformer (`glow.transform()`).

Combining the functionality listed above with the standard Spark functionality, researchers can read in large genomic data files, transform them, run bioinformatics pipelines at scale, and even prepare and run machine learning analysis.

By having your genomics data in a data lake, Azure Databricks can easily connect to it and retrieve data for scalable analyses using the Runtime for Genomics, Glow, and many other open-source libraries, all in an interactive and collaborative notebook-style environment.

To learn more about the Azure Databricks Runtime for Genomics and Glow, visit docs.microsoft.com/en-us/azure/databricks/applications/genomics/.

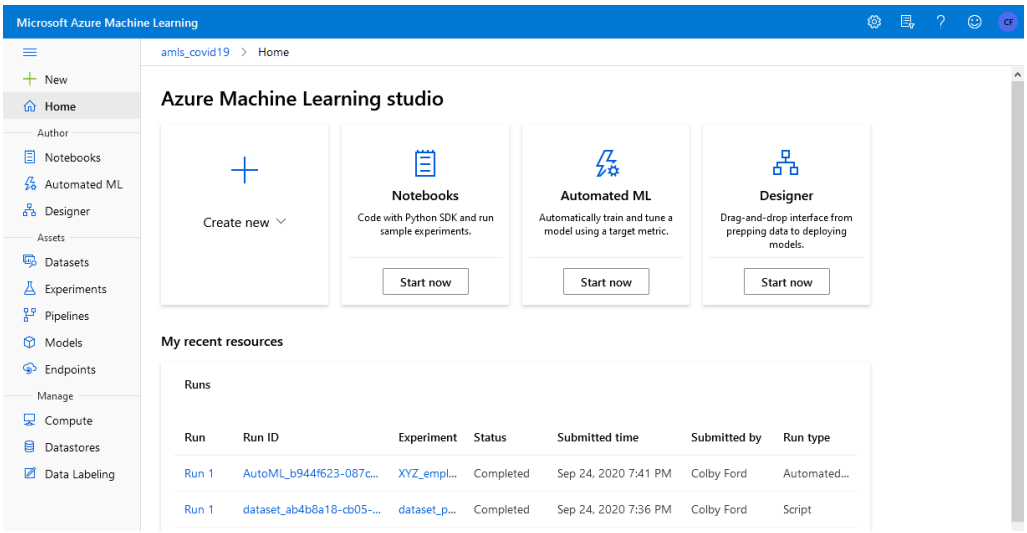
Azure Machine Learning

Azure Machine Learning is an enterprise-grade service for performing machine learning in the cloud. This service offers an excellent interface for machine learning experimentation, dataset management, model deployment, and model management. Plus, this service also provides a visual designer and automated machine learning for code-free predictive analytics. Azure Machine Learning has both a Python SDK and an R SDK, so you can integrate your machine learning process with this service and allow it to track model performance and aid in the operationalization of models as REST APIs.

Despite its name, Azure Machine Learning is an excellent tool for use in bioinformatics as it provides web-based versions of RStudio Server and Jupyter, which are commonly used in the field. This is done through creating a "compute instance", which is simply a managed cloud-based workstation that comes with all sorts of relevant software pre-installed.

With RStudio, R programmers can use virtually any open-source package that they know and love (and this includes Bioconductor packages, as well). Similarly, Python lovers can work in JuptyerLab or a traditional Juptyer notebook. This makes this service an excellent add on to use with a genomics data lake as your team can easily connect to data in the same Azure environment and perform their bioinformatics analyses in the cloud, all while using familiar tools.

To learn more about Azure Machine Learning, visit azure.microsoft.com/en-us/services/machine-learning/.

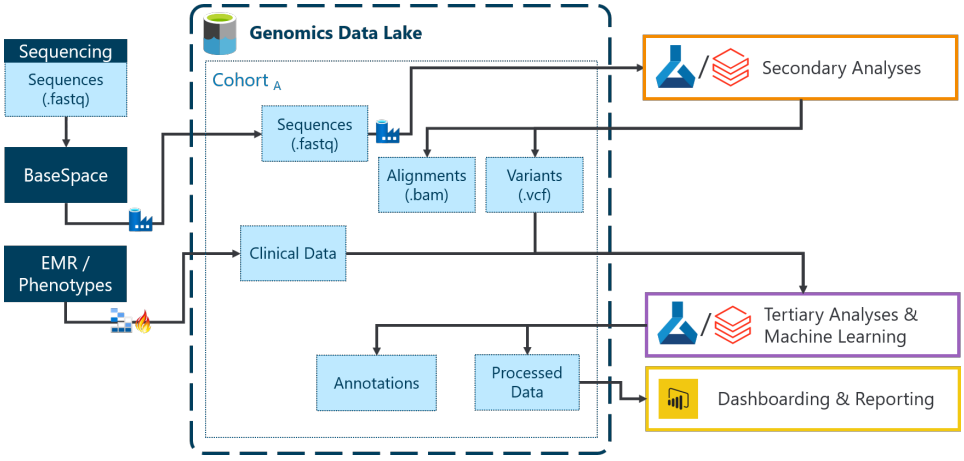


Why Azure?

If you are used to using your local workstation to manually run bioinformatics pipelines on your genomics data, you know how slow this process can be if you do not have adequate compute resources on that machine.

One option is to scale your processing by placing the workload on your institution's High-Performance Computing (HPC) environment. This option often limits your ability to interactively develop in that environment and you now have to wait on the cluster's pesky scheduler (like Torque or Slurm) to actually run your job. Plus, for some organizations, running complex pipelines in an automated way is quite difficult or not supported given the limitations around on-premise architecture. As a bioinformatician, your time is better spent doing research rather than executing pipelines by hand.

The Azure cloud alleviates many of these issues, offering almost limitless compute power all under your control. With the services we have covered in this book, your organization can easily organize data in a genomics data lake, orchestrate and automate its movement between locations and services using Azure Data Factory, and analyze it at scale using services like Azure Machine Learning or Azure Databricks.



Start using Azure for free today by visiting azure.microsoft.com/en-us/free/.

Bibliography

[1] International Human Genome Sequencing Consortium Publishes Sequence and Analysis of the Human Genome, Feb 2001. URL www.genome.gov/10002192/2001-release-first-analysis-of-human-genome.

[2] Zachary D. Stephens, Skylar Y. Lee, Faraz Faghri, Roy H. Campbell, Chengxiang Zhai, Miles J. Efron, Ravishankar Iyer, Michael C. Schatz, Saurabh Sinha, and Gene E. Robinson. Big data: Astronomical or genomical? *PLOS Biology*, 13(7):1–11, 07 2015. doi: 10.1371/journal.pbio.1002195. URL https://doi.org/10.1371/journal.pbio.1002195.

[3] Roy Williams. Data powers of ten. URL https://cs.calvin.edu/courses/is/341/private/powers.html.

[4] Michael Armbrust, Tathagata Das, Liwen Sun, Burak Yavuz, Shixiong Zhu, Mukul Murthy, Joseph Torres, Herman van Hovell, Adrian Ionescu, Alicja undefineduszczak, Michał undefinedwitakowski, Michał Szafrański, Xiao Li, Takuya Ueshin, Mostafa Mokhtar, Peter Boncz, Ali Ghodsi, Sameer Paranjpye, Pieter Senster, Reynold Xin, and Matei Zaharia. Delta lake: High-performance acid table storage over cloud object stores. 13(12):3411–3424, August 2020. ISSN 2150-8097. doi: 10.14778/3415478.3415560. URL https://doi.org/10.14778/3415478.3415560.

[5] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. Apache spark: A unified engine for big data processing. *Commun. ACM*, 59(11):56–65, October 2016. ISSN 0001-0782. doi: 10.1145/2934664. URL https://doi.org/10.1145/2934664.

[6] P. Cingolani, A. Platts, M. Coon, T. Nguyen, L. Wang, S.J. Land, X. Lu, and D.M. Ruden. A program for annotating and predicting the effects of single nucleotide polymorphisms, snpeff: Snps in the genome of drosophila melanogaster strain w1118; iso-2; iso-3. *Fly*, 6(2):80–92, 2012.

[7] William McLaren, Laurent Gil, Sarah E. Hunt, Harpreet Singh Riat, Graham R. S. Ritchie, Anja Thormann, Paul Flicek, and Fiona Cunningham. The ensembl variant effect predictor. *Genome Biology*, 17(1):122, Jun 2016. ISSN 1474-760X. doi: 10.1186/s13059-016-0974-4. URL https://doi.org/10.1186/s13059-016-0974-4.

[8] Hail library. URL https://github.com/hail-is/hail.

[9] Matt Massie, Frank Nothaft, Christopher Hartl, Christos Kozanitis, André Schumacher, Anthony D Joseph, and David A Patterson. ADAM: Genomics formats and processing patterns for cloud scale computing. Technical report, UCB/EECS-2013-207, EECS Department, University of California, Berkeley, 2013.

[10] Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, and Mark A. DePristo. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 20(9):1297–1303, 2010. doi: 10.1101/gr.107524.110.

[11] Wei Zhou, Zhangchen Zhao, Jonas B. Nielsen, Lars G. Fritsche, Jonathon LeFaive, Sarah A. Gagliano Taliun, Wenjian Bi, Maiken E. Gabrielsen, Mark J. Daly, Benjamin M. Neale, Kristian Hveem, Goncalo R. Abecasis, Cristen J. Willer, and Seunggeun Lee. Scalable generalized linear mixed model for region-based association tests in large biobanks and cohorts. *Nature Genetics*, 52(6):634–639, Jun 2020. ISSN 1546-1718. doi: 10.1038/s41588-020-0621-6. URL https://doi.org/10.1038/s41588-020-0621-6.

[12] Karen Feng, Henry Davidge, Kiavash Kianfar, William Brandler, Ahir Reddy, Amir Kermany, Boris Boutkov, Frank Austin Nothaft, Herman van Hovell, Leland, and et al. projectglow/glow: v0.6.0. Sep 2020. doi: 10.5281/zenodo.4022964.

[13] Joelle Mbatchou, Leland Barnard, Joshua Backman, Anthony Marcketta, Jack A. Kosmicki, Andrey Ziyatdinov, Christian Benner, Colm O'Dushlaine, Mathew Barber, Boris Boutkov, Lukas Habegger, Manuel Ferreira, Aris Baras, Jeffrey Reid, Gonçalo Abecasis, Evan Maxwell, and Jonathan Marchini. Computationally efficient whole genome regression for quantitative and binary traits. *bioRxiv*, 2020. doi: 10.1101/2020.06.19.162354. URL https://www.biorxiv.org/content/early/2020/06/22/2020.06.19.162354.

How to Cite

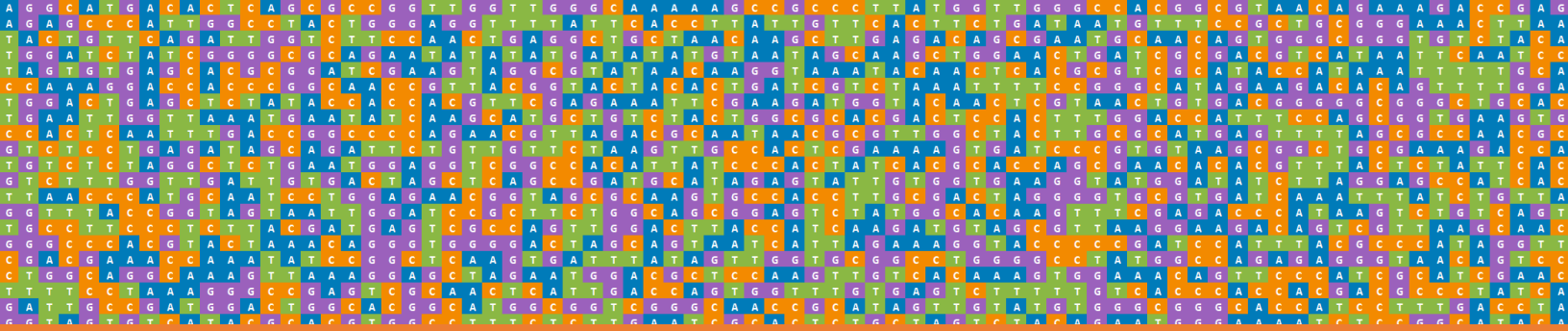
If you reference material from this book in your research, please cite us:

Generic Text Version:

Ford, Colby T. and Baker, Larry. (2021). Building a Genomics Data Lake in Azure. BlueGranite, Inc. doi: 10.5281/zenodo.4474520. URL https://www.bluegranite.com/genomics-data-lake-ebook

BibTeX **Version:**

```
@book{bluegranite_genomics,
  author={Ford, Colby T. and Baker, Larry},
  title={{Building a Genomics Data Lake in Azure}},
  publisher={BlueGranite, Inc.},
  year={2021},
  month={Jan},
  DOI={10.5281/zenodo.4474520},
  URL={https://www.bluegranite.com/genomics-data-lake-ebook}
  }
```



BlueGranite.com/Genomics

BLUEGRANITE

DATA • INSIGHTS • ANALYTICS