**Project Ideas**

|  |
| --- |
|  |
| |  | | --- | | **Exploratory data analysis of genomic datasets using ADAM and Mango with Apache Spark on Amazon EMR**  13 JUL 2018  Amazon EMR, AWS Big Data  <https://aws.amazon.com/blogs/big-data/exploratory-data-analysis-of-genomic-datasets-using-adam-and-mango-with-apache-spark-on-amazon-emr/> | | As the cost of genomic sequencing has rapidly decreased, the amount of publicly available genomic data has soared over the past couple years. New cohorts and studies have produced massive datasets consisting of over 100,000 individuals. Simultaneously, these datasets have been processed to extract genetic variation across populations, producing mass amounts of variation data for each cohort. In this era of big data, tools like Apache Spark have provided a user-friendly platform for batch processing large datasets. However, in order to use such tools as a sufficient replacement to current bioinformatics pipelines, we need more accessible and comprehensive API’s for processing genomic data, as well as support for interactive exploration of these processed datasets.  [**ADAM**](https://github.com/bigdatagenomics/adam)  <https://github.com/bigdatagenomics/adam>  and  [**Mango**](https://github.com/bigdatagenomics/mango)  <https://github.com/bigdatagenomics/mango>  provide a unified environment for processing, filtering, and visualizing large genomic datasets on Apache Spark. ADAM allows users to programmatically load, process, and select raw genomic and variation data using SparkSQL, an SQL interface for aggregating and selecting data in Apache Spark. Mango supports visualization of both raw and aggregated genomic data in a Jupyter notebook environment, allowing users to draw conclusions from large datasets at multiple resolutions. This combined power of ADAM and Mango allows users to load, query and explore datasets in a unified environment, allowing users to interactively explore genomic data at a scale previously impossible using single node bioinformatics tools. **Configuring ADAM and Mango on Amazon EMR** First, we will launch and configure an EMR cluster. Mango uses Docker containers to easily run on Amazon EMR. Upon cluster startup, EMR will use the bootstrap action below to install Docker and the required startup scripts. The scripts will be available at /home/hadoop/mango-scripts  aws emr create-cluster  --release-label emr-5.19.0 \  --name 'emr-5.19.0 Mango example' \  --applications Name=Hadoop Name=Hive Name=Spark \  --ec2-attributes KeyName=<your-ec2-key>,InstanceProfile=EMR\_EC2\_DefaultRole \  --service-role EMR\_DefaultRole \  --instance-groups \ InstanceGroupType=MASTER,InstanceCount=1,InstanceType=c5.4xlarge \ InstanceGroupType=CORE,InstanceCount=4,InstanceType=c5.4xlarge \ --region <your-aws-region> \  --log-uri s3://<your-s3-bucket>/emr-logs/ \  --bootstrap-actions \  Name='Install Mango', Path="s3://aws-bigdata-blog/artifacts/mango-emr/install-bdg-mango-docker-emr5.sh"  To start the Mango notebook, run the following:  /home/hadoop/mango-scripts/run-notebook.sh  This file will set up all of the environment variables needed to run Mango in Docker on EMR. In your terminal, you will see the port and Jupyter notebook token for the Mango Notebook session. Navigate to this port on the public DNS URL of the master node for your EMR cluster. **Loading data from the 1000 Genomes Project** Now that we have a working environment, lets use ADAM and Mango to discover interesting variants in the child from the genome sequencing data of a trio (data from a mother, father, and child). These data are available from the  [1000 Genomes Project AWS Public Dataset](https://aws.amazon.com/1000genomes/).  <https://registry.opendata.aws/1000-genomes/>  **Resources on AWS**  Description  <http://www.internationalgenome.org/formats>  Resource type  S3 Bucket  Amazon Resource Name (ARN)  arn:aws:s3:::1000genomes  AWS Region  us-east-1  AWS CLI Access (No AWS account required)  aws s3  In this analysis, we will view a trio ([NA19685](http://www.internationalgenome.org/data-portal/sample/NA19685), [NA19661](http://www.internationalgenome.org/data-portal/sample/NA19661), and [NA19660](http://www.internationalgenome.org/data-portal/sample/NA19660)) and search for variants that are present in the child but not present in the parents.  In particular, we want to identify genetic variants that are found in the child but not in the parents, known as de novo variants. These are interesting regions, as they may indicate sights of de novo variation that may contribute to multiple disorders.  You can find the Jupyter notebook containing these examples in [Mango’s GitHub repository](https://github.com/bigdatagenomics/mango/blob/master/example-files/notebooks/aws-1000genomes.ipynb), or at /opt/cgl-docker-lib/mango/example-files/notebooks/aws-1000genomes.ipynb in the running Docker container for Mango.  First, import the ADAM and Mango modules and any Spark modules that you need:  # Import ADAM modules  from bdgenomics.adam.adamContext import ADAMContext  from bdgenomics.adam.rdd import AlignmentRecordRDD, CoverageRDD  from bdgenomics.adam.stringency import LENIENT, \_toJava  # Import Mango modules  from bdgenomics.mango.rdd import GenomicVizRDD  from bdgenomics.mango.QC import CoverageDistribution  # Import Spark modules  from pyspark.sql import functions as sf  Next, create a Spark session. You will use this session to run SQL queries on variants.  # Create ADAM Context  ac = ADAMContext(spark)  **Variant analysis with Spark SQL** Load in a subset of variant data from chromosome 17:  genotypesPath = 's3://1000genomes/phase1/analysis\_results/integrated\_call\_sets/ALL.chr17.integrated\_phase1\_v3.20101123.snps\_indels\_svs.genotypes.vcf.gz'  genotypes = ac.loadGenotypes(genotypesPath)  # repartition genotypes to balance the load across memory  genotypes\_df = genotypes.toDF()  You can take a look at the schema by printing the columns in the dataframe.  # cache genotypes and show the schema  genotypes\_df.columns  This genotypes dataset contains all samples from the 1000 Genomes Project. Therefore, you will next filter genotypes to only consider samples that are in the NA19685 trio, and cache the results in memory.  # trio IDs  IDs = ['NA19685', 'NA19661','NA19660']  # Filter by individuals in the trio  trio\_df = genotypes\_df.filter(genotypes\_df["sampleId"].isin(IDs))  trio\_df.cache()  trio\_df.count()  Next, add a new column to your dataframe that determines the genomic location of each variant. This is defined by the chromosome (contigName) and the start and end position of the variant.  # Add ReferenceRegion column and group by referenceRegion  trios\_with\_referenceRegion = trio\_df.withColumn('ReferenceRegion',  sf.concat(sf.col('contigName'),sf.lit(':'), sf.col('start'), sf.lit('-'), sf.col('end')))  Now, you can query your dataset to find de novo variants. But first, you must register your dataframe with Spark SQL.  # Register df with Spark SQL  trios\_with\_referenceRegion.createOrReplaceTempView("trios")  Now that your dataframe is registered, you can run SQL queries on it. For the first query, select the names of variants belonging to sample NA19685 that have at least one alternative (ALT) allele.  # filter by alleles. This is a list of variant names that have an alternate allele for the child  alternate\_variant\_sites = spark.sql("SELECT variant.names[0] AS snp FROM trios \  WHERE array\_contains(alleles, 'ALT') AND sampleId == 'NA19685'")  collected\_sites = list(map(lambda x: x.snp, alternate\_variant\_sites.collect())  For your next query, filter sites in which the parents have both reference alleles. Then filter these variants by the set produced previously from the child.  # get parent records and filter by only REF locations for variant names that were found in the child with an ALT  filtered1 = spark.sql("SELECT \* FROM trios WHERE sampleId == 'NA19661' or sampleId == 'NA19660' \  AND !array\_contains(alleles, 'ALT')")  filtered2 = filtered1.filter(filtered1["variant.names"][0].isin(collected\_sites))  snp\_counts = filtered2.groupBy("variant.names").count().collect()  # collect snp names as a list  snp\_names = map(lambda x: x.names, snp\_counts)  denovo\_snps = [item for sublist in snp\_names for item in sublist]  denovo\_snps[:10]    Now that you have found some interesting variants, you can unpersist your genotypes from memory.  trio\_df.unpersist()  **Working with alignment data** You have found a lot of potential de novo variant sites. Next, you can visually verify some of these sites to see if the raw alignments match up with these de novo hits.  First, load in the alignment data for the NA19685 trio:  # load in NA19685 exome from s3a  childReadsPath = 's3a://1000genomes/phase1/data/NA19685/exome\_alignment/NA19685.mapped.illumina.mosaik.MXL.exome.20110411.bam'  parent1ReadsPath = 's3a://1000genomes/phase1/data/NA19685/exome\_alignment/NA19660.mapped.illumina.mosaik.MXL.exome.20110411.bam'  parent2ReadsPath = 's3a://1000genomes/phase1/data/NA19685/exome\_alignment/NA19661.mapped.illumina.mosaik.MXL.exome.20110411.bam'  childReads = ac.loadAlignments(childReadsPath, stringency=LENIENT)  parent1Reads = ac.loadAlignments(parent1ReadsPath, stringency=LENIENT)  parent2Reads = ac.loadAlignments(parent2ReadsPath, stringency=LENIENT)  Note that this example uses s3a:// instead of s3:// style URLs. The reason for this is that the ADAM formats use Java NIO to access BAM files. To do this, we are using a [JSR 203](https://jcp.org/en/jsr/detail?id=203) implementation for the Hadoop Distributed File System to access these files. This itself requires the s3a:// protocol. You can view that implementation in [this GitHub repository](https://github.com/fnothaft/jsr203-s3a).  You now have data alignment data for three individuals in your trio. However, the data has not yet been loaded into memory. To cache these datasets for fast subsequent access to the data, run the cache() function:  # cache child RDD and count records  # takes about 2 minutes, on 4 c3.4xlarge worker nodes  childReads.cache()  # Count reads in the child  childReads.toDF().count()  # Output should be 95634679  **Quality control of alignment data** One popular analysis to visually re-affirm the quality of genomic alignment data is by viewing coverage distribution. Coverage distribution gives you an idea of the read coverage that you have across a sample.  Next, generate a sample coverage distribution plot for the child alignment data on chromosome 17:  # Calculate read coverage  # Takes 2-3 minutes  childCoverage = childReads.transform(lambda x: x.filter(x.contigName == "17")).toCoverage()  childCoverage.cache()  childCoverage.toDF().count()  # Output should be 51252612  Now that coverage data is calculated and cached, compute the coverage distribution of chromosome 17 and plot the coverage distribution:  # Calculate coverage distribution  # You can check the progress in the SparkUI by navigating to  # :8088 and clicking on the currently running Spark application.  cd = CoverageDistribution(spark, childCoverage, bin\_size = 1)  ax, results = cd.plotDistributions(normalize=True, cumulative=False)  ax.set\_title("Normalized Target Region Coverage")  ax.set\_ylabel("Fraction")  ax.set\_xlabel("Coverage Depth")  ax.set\_xscale("log")  plt.show()      This looks pretty standard because the data you are viewing is exome data. Therefore, you can see a high number of sights with low coverage and a smaller number of genomic positions with more than 100 reads. Now that you are done with coverage, you can unpersist these datasets to clear space in memory for the next analysis.  childCoverage.unpersist()  **Viewing sites with missense variants in the proband** After verifying alignment data and filtering variants, you have four genes with potential missense mutations in the proband, including YBX2, ZNF286B, KSR1, and GNA13. You can visually verify these sites by filtering and viewing the raw reads of the child and parents.  First, view the child reads. If you zoom in to the location of the GNA13 variant (63052580-63052581), you can see a heterozygous T to A call:  # missense variant at GNA13: 63052580-63052581 (SNP rs201316886)  # define alignment summary for child reads  childViz = AlignmentSummary(spark, ac, childReads)  # Takes about 2 minutes to collect data from workers  contig = "17"  start = 63052180  end = 63052981  childViz.viewPileup(contig, start, end)    It looks like there indeed is a variant at this position, possibly a heterozygous SNP with alternate allele A. Look at the parent data to verify that this variant does not appear in the parents:  # define alignment summary for parent reads  parent1Viz = AlignmentSummary(spark, ac, parent1Reads)  # view missense variant at GNA13: 63052580-63052581 in parent 1  contig = "17"  start = 63052180  end = 63052981  parent1Viz.viewPileup(contig, start, end)    This confirms the filter that this variant is indeed present only in the proband, but not the parents. **Summary** To summarize, this post demonstrated how to set up and run ADAM and Mango in Amazon EMR. We demonstrated how to use these tools in an interactive notebook environment to explore the 1000 Genomes dataset, a publicly available dataset on Amazon S3. We used these tools inspect 1000 Genomes data quality, query for interesting variants in the genome, and validate results through the visualization of raw data.  For more information about Mango, see the [Mango User Guide](http://bdg-mango.readthedocs.io/en/latest/). If you have questions or suggestions, please comment below. |  |  | | --- | | **Orchestrating analytics jobs by running Amazon EMR Notebooks programmatically**  23 NOV 2020 | Amazon EMR, AWS Big Data  <https://aws.amazon.com/blogs/big-data/orchestrating-analytics-jobs-by-running-amazon-emr-notebooks-programmatically/> | | [Amazon EMR](http://aws.amazon.com/emr) is a big data service offered by AWS to run Apache Spark and other open-source applications on AWS in a cost-effective manner. [Amazon EMR Notebooks](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-managed-notebooks.html) is a managed environment based on [Jupyter Notebook](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Notebook) that allows data scientists, analysts, and developers to prepare and visualize data, collaborate with peers, build applications, and perform interactive analysis using EMR clusters.  EMR notebook APIs are available on Amazon EMR release version 5.18.0 or later and can be used to run EMR notebooks via a script or command line. The ability to start, stop, list, and describe EMR notebook runs without the Amazon EMR console enables you to programmatically control running an EMR notebook. Using a parameterized notebook cell allows you to pass different parameter values to a notebook without having to create a copy of the notebook for each new set of parameter values. With this feature, you can schedule running EMR notebooks with cron scripts, chain multiple EMR notebooks, and use orchestration services such as [AWS Step Functions](https://aws.amazon.com/step-functions/) or Apache Airflow to build pipelines. If you want to use EMR notebooks in a non-interactive manner, this enables you to run ETL workloads, especially in production.  In this post, we show how to orchestrate analytics jobs by running EMR Notebooks programmatically with the following two use cases:   * Scheduling an EMR notebook run via crontab and the [AWS Command Line Interface](http://aws.amazon.com/cli) (AWS CLI) * Chaining your notebooks with Step Functions triggered by [Amazon CloudWatch Events](https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/WhatIsCloudWatchEvents.html)   For our data source, we use the open-source, real-time COVID-19 US daily case reports provided by Johns Hopkins University CSSE in the following [GitHub repo](https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_daily_reports_us). Prerequisites Before getting started, you must have the following prerequisites:   * An AWS account that provides access to the following AWS services at least:   + [AWS CloudFormation](http://aws.amazon.com/cloudformation)   + Amazon CloudWatch   + [Amazon Elastic Compute Cloud](http://aws.amazon.com/ec2) (Amazon EC2)   + Amazon EMR   + [Amazon EventBridge](https://aws.amazon.com/eventbridge/)   + [AWS Identity and Access Management](http://aws.amazon.com/iam) (IAM)   + [AWS Lambda](http://aws.amazon.com/lambda)   + [Amazon Simple Storage Service](http://aws.amazon.com/s3) (Amazon S3)   + AWS Step Functions * AWS CLI Version 1.18.128 or later installed on your work station. * [Jupyter](https://jupyter.org/install) installed on your work station (this is used for the output visualization part for this post only). * An EMR cluster running Amazon EMR release 5.18.0 or later, with Hadoop, Spark, and Livy installed. Record the value of the cluster ID (for example, <j-\*\*\*\*\*\*\*\*\*\*\*\*\*>); you use this for the examples later. * An EMR notebook created on the Amazon EMR console, using the following two input notebook files:   + [demo\_pyspark.pynb](https://aws-bigdata-blog.s3.amazonaws.com/artifacts/aws-blog-runnable-notebook/demo/notebook/demo_pyspark.ipynb) – Used for both use cases in this post.   + [trailing\_N day.ipynb](https://aws-bigdata-blog.s3.amazonaws.com/artifacts/aws-blog-runnable-notebook/demo/notebook/trailing_N_day.ipynb) – Used for the second use case.   Record the notebook ID (for example, <e-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*>); you use this later for our examples later. Organize the notebook files in the Jupyter UI as follows:   * /demo\_pyspark.ipynb * /experiment/trailing\_N\_day.ipynb   See [Creating a Notebook](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-managed-notebooks-create.html) for more information on how to create an EMR notebook. |  |  | | --- | | **Interactive Analysis of Genomic Datasets Using Amazon Athena**  by Aaron Friedman, PhD | on 07 DEC 2016 | in Amazon Athena, AWS Big Data  [**https://aws.amazon.com/blogs/big-data/interactive-analysis-of-genomic-datasets-using-amazon-athena/**](https://aws.amazon.com/blogs/big-data/interactive-analysis-of-genomic-datasets-using-amazon-athena/) | | Aaron Friedman is a Healthcare and Life Sciences Solutions Architect with Amazon Web Services  The genomics industry is in the midst of a data explosion. Due to the rapid drop in the cost to sequence genomes, genomics is now central to many medical advances. When your genome is sequenced and analyzed, raw sequencing files are processed in a multi-step workflow to identify where your genome differs from a standard reference. Your variations are stored in a Variant Call Format (VCF) file, which is then combined with other individuals to enable population-scale analyses. Many of these datasets are publicly available, and an increasing number are hosted on AWS as part of our [Open Data](https://aws.amazon.com/public-data-sets/) project.  To mine genomic data for new discoveries, researchers in both industry and academia build complex models to analyze populations at scale. When building models, they first explore the datasets-of-interest to understand what questions the data might answer. In this step, interactivity is key, as it allows them to move easily from one question to the next.  Recently, we launched [Amazon Athena](https://aws.amazon.com/athena/) as an interactive query service to analyze data on [Amazon S3](https://aws.amazon.com/s3/). With Amazon Athena there are no clusters to manage and tune, no infrastructure to setup or manage, and customers pay only for the queries they run. Athena is able to query many file types straight from S3. This flexibility gives you the ability to interact easily with your datasets, whether they are in a raw text format (CSV/JSON) or specialized formats (e.g. Parquet). By being able to flexibly query different types of data sources, researchers can more rapidly progress through the data exploration phase for discovery. Additionally, researchers don’t have to know nuances of managing and running a big data system. This makes Athena an excellent complement to data warehousing on [Amazon Redshift](https://aws.amazon.com/redshift/) and big data analytics on [Amazon EMR](https://aws.amazon.com/emr/).   In this post, I discuss how to prepare genomic data for analysis with Amazon Athena as well as demonstrating how Athena is well-adapted to address common genomics query paradigms.  I use the [Thousand Genomes dataset](https://aws.amazon.com/1000genomes/) hosted on Amazon S3, a seminal genomics study, to demonstrate these approaches. All code that is used as part of this post is available in our GitHub [repository](https://github.com/awslabs/aws-big-data-blog/tree/master/aws-blog-athena-genomics/).  Although this post is focused on genomic analysis, similar approaches can be applied to any discipline where large-scale, interactive analysis is required.   Select, aggregate, annotate query pattern in genomics Genomics researchers may ask different questions of their dataset, such as:   * What variations in a genome may increase the risk of developing disease? * What positions in the genome have abnormal levels of variation, suggesting issues in quality of sequencing or errors in the genomic reference? * What variations in a genome influence how an individual may respond to a specific drug treatment? * Does a group of individuals contain a higher frequency of a genomic variant known to alter response to a drug relative to the general population?   All these questions, and more, can be generalized under a common query pattern I like to call “Select, Aggregate, Annotate”. Some of our genomics customers, such as Human Longevity, Inc., routinely use this query pattern [in their work](https://www.youtube.com/watch?v=CGbWEkszAlQ).  In each of the above queries, you execute the following steps:  **SELECT:** Specify the cohort of individuals meeting certain criteria (disease, drug response, age, BMI, entire population, etc.).  **AGGREGATE:** Generate summary statistics of genomic variants across the cohort that you selected.  **ANNOTATE:** Assign meaning to each of the variants by joining on known information about each variant. Dataset preparation Properly organizing your dataset is one of the most critical decisions for enabling fast, interactive analyses. Based on the query pattern I just described, the table representing your population needs to have the following information:   * A unique sample ID corresponding to each sample in your population * Information about each variant, specifically its location in the genome as well as the specific deviation from the reference * Information about how many times in a sample a variant occurs (0, 1, or 2 times) as well as if there are multiple variants in the same site. This is known as a genotype.   The extract, transform, load (ETL) process to generate the appropriate data representation has two main steps. First, you use ADAM, a genomics analysis platform built on top of Spark, to convert the variant information residing a VCF file to Parquet for easier downstream analytics, in a process similar to the one described in the [Will Spark Power the Data behind Precision Medicine?](https://blogs.aws.amazon.com/bigdata/post/Tx1GE3J0NATVJ39/Will-Spark-Power-the-Data-behind-Precision-Medicine) post. Then, you use custom Python code to massage the data and select only the appropriate fields that you need for analysis with Athena.  First, [spin up an EMR cluster](http://docs.aws.amazon.com/ElasticMapReduce/latest/ManagementGuide/emr-gs.html) (version 5.0.3) for the ETL process. I used a c4.xlarge for my master node and m4.4xlarges with 1 TB of scratch for my core nodes.  After you SSH into your master node, clone the [git repository](https://github.com/awslabs/aws-big-data-blog.git). You can also put this in as a bootstrap action when spinning up your cluster. |  |  | | --- | | **Building High-Throughput Genomics Batch Workflows on AWS: Introduction (Part 1 of 4)**  <https://aws.amazon.com/blogs/compute/building-high-throughput-genomics-batch-workflows-on-aws-introduction-part-1-of-4/>  by Andy Katz | on 30 MAY 2017 | | | **Genomics Research on AWS**  <https://github.com/aws-samples/aws-batch-genomics/tree/v1.0.0>  A tutorial on how to package and deploy a bioinformatics workflow on AWS using AWS Batch  This tutorial will cover the material presented within the "Genomics Workflows on AWS" blog post series  (Part 1, Part 2, Part 3, Part 4) that covers the basics of bootstrapping a bioinformatics analysis pipeline on AWS.  We break down the tutorial roughly as follows:  Setting up your AWS account (if you do not already have one)  Package a set of bioinformatics applications using Docker  Create a AWS Batch environment for analysis  Define and deploy AWS Step Functions to control the data processing steps  Initiate a workflow |  |  | | --- | | **Optimizing for cost, availability and throughput by selecting your AWS Batch allocation strategy**  by Bala Thekkedath | on 24 OCT 2019 | in Advanced (300), Amazon EC2, AWS Batch  <https://aws.amazon.com/blogs/compute/optimizing-for-cost-availability-and-throughput-by-selecting-your-aws-batch-allocation-strategy/> | | AWS offers a broad range of instances that are advantageous for batch workloads. The scale and provisioning speed of AWS’ compute instances allow you to get up and running at peak capacity in minutes without paying for downtime. Today, I’m pleased to introduce allocation strategies: a significant new capability in [AWS Batch](https://docs.aws.amazon.com/batch/latest/userguide/allocation-strategies.html) that  makes provisioning compute resources flexible and simple. In this blog post, I explain how the AWS Batch allocation strategies work, when you should use them for your workload, and provide an example CloudFormation script. This blog helps you get started on building your personalized Compute Environment (CE) most appropriate to your workloads.  **Overview**  AWS Batch is a fully managed, cloud-native batch scheduler. It manages the queuing and scheduling of your batch jobs, and the resources required to run your jobs. One of AWS Batch’s great strengths is the ability to manage instance provisioning as your workload requirements and budget needs change. AWS Batch takes advantage of AWS’s broad base of compute types. For example, you can launch compute based instances and memory instances that can handle different workload types, without having to worry about building a cluster to meet peak demand.  Previously, AWS Batch had a cost-controlling approach to manage compute instances for your workloads. The service chose an instance that was the best fit for your jobs based on vCPU, memory, and GPU requirements, at the lowest cost. Now, the newly added allocation strategies provide flexibility. They allow AWS Batch to consider capacity and throughput in addition to cost when provisioning your instances. This allows you to leverage different priorities when launching instances depending on your workloads’ needs, such as: controlling cost, maximizing throughput, or minimizing [Amazon EC2 Spot](https://aws.amazon.com/ec2/spot/) instances interruption rates.  There are now three instance allocation strategies from which to choose when creating an AWS Batch [Compute Environment](https://docs.aws.amazon.com/batch/latest/userguide/compute_environments.html) (CE). They are:  1.        Spot Capacity Optimized  2.        Best Fit Progressive  3.        Best Fit |  |  | | --- | | **BioContainers**  [**https://biocontainers-edu.readthedocs.io/en/latest/**](https://biocontainers-edu.readthedocs.io/en/latest/)  **Integration with BioConda**  [**https://biocontainers-edu.readthedocs.io/en/latest/conda\_integration.html**](https://biocontainers-edu.readthedocs.io/en/latest/conda_integration.html)  **bioconda / packages  View all (8203)** [**https://anaconda.org/bioconda/**](https://anaconda.org/bioconda/)  [**https://anaconda.org/bioconda/repo**](https://anaconda.org/bioconda/repo) | | **Bioconda**  <https://bioconda.github.io/>  **Bioconda** is a channel for the [conda](https://conda.io/en/latest/index.html) package manager specializing in bioinformatics software. Bioconda consists of:   * a [repository of recipes](https://github.com/bioconda/bioconda-recipes) hosted on GitHub * a [build system](https://github.com/bioconda/bioconda-utils) turning these recipes into conda packages * a [repository of packages](https://anaconda.org/bioconda/) containing over 7000 bioinformatics packages ready to use with conda install * over 850 contributors and 570 members who add, modify, update and maintain the recipes   The conda package manager makes installing software a vastly more streamlined process. Conda is a combination of other package managers you may have encountered, such as pip, CPAN, CRAN, Bioconductor, apt-get, and homebrew. Conda is both language- and OS-agnostic, and can be used to install C/C++, Fortran, Go, R, Python, Java etc programs on Linux, Mac OSX, and Windows.  Conda allows separation of packages into repositories, or channels. The main defaults channel has a large number of common packages. Users can add additional channels from which to install software packages not available in the defaults channel. Bioconda is one such channel specializing in bioinformatics software.  **Browse packages in the Bioconda channel:** [Package Index](https://bioconda.github.io/conda-package_index.html)  Each package added to Bioconda also has a corresponding Docker [BioContainer](https://biocontainers.pro) automatically created and uploaded to [Quay.io](https://quay.io/organization/biocontainers). A list of these and other containers can be found at the [Biocontainers Registry](https://biocontainers.pro/#/registry). | | **bioconda / packages / samtools 1.11**  <https://anaconda.org/bioconda/samtools>  Tools for dealing with SAM, BAM and CRAM files  **bioconda / packages / igv 2.8.13**  <https://anaconda.org/bioconda/igv>  Integrative Genomics Viewer. Fast, efficient, scalable visualization tool for genomics data and annotations.  **bioconda / packages / bioconductor-deseq2 1.30.0**  <https://anaconda.org/bioconda/bioconductor-deseq2>  Differential gene expression analysis based on the negative binomial distribution  **bioconda / packages / bioconductor-pcaexplorer 2.16.0**  <https://anaconda.org/bioconda/bioconductor-pcaexplorer>  Interactive Visualization of RNA-seq Data Using a Principal Components Approach | |

|  |
| --- |
|  |
| |  | | --- | | **Data Science (Python): Project Repositories** | |  |  |  | | --- | | Optimizing for cost, availability and throughput by selecting your AWS Batch allocation strategy  by Bala Thekkedath | on 24 OCT 2019 | in Advanced (300), Amazon EC2, AWS Batch  <https://aws.amazon.com/blogs/compute/optimizing-for-cost-availability-and-throughput-by-selecting-your-aws-batch-allocation-strategy/> | | AWS offers a broad range of instances that are advantageous for batch workloads. The scale and provisioning speed of AWS’ compute instances allow you to get up and running at peak capacity in minutes without paying for downtime. Today, I’m pleased to introduce allocation strategies: a significant new capability in [AWS Batch](https://docs.aws.amazon.com/batch/latest/userguide/allocation-strategies.html) that  makes provisioning compute resources flexible and simple. In this blog post, I explain how the AWS Batch allocation strategies work, when you should use them for your workload, and provide an example CloudFormation script. This blog helps you get started on building your personalized Compute Environment (CE) most appropriate to your workloads.  **Overview**  AWS Batch is a fully managed, cloud-native batch scheduler. It manages the queuing and scheduling of your batch jobs, and the resources required to run your jobs. One of AWS Batch’s great strengths is the ability to manage instance provisioning as your workload requirements and budget needs change. AWS Batch takes advantage of AWS’s broad base of compute types. For example, you can launch compute based instances and memory instances that can handle different workload types, without having to worry about building a cluster to meet peak demand.  Previously, AWS Batch had a cost-controlling approach to manage compute instances for your workloads. The service chose an instance that was the best fit for your jobs based on vCPU, memory, and GPU requirements, at the lowest cost. Now, the newly added allocation strategies provide flexibility. They allow AWS Batch to consider capacity and throughput in addition to cost when provisioning your instances. This allows you to leverage different priorities when launching instances depending on your workloads’ needs, such as: controlling cost, maximizing throughput, or minimizing [Amazon EC2 Spot](https://aws.amazon.com/ec2/spot/) instances interruption rates.  There are now three instance allocation strategies from which to choose when creating an AWS Batch [Compute Environment](https://docs.aws.amazon.com/batch/latest/userguide/compute_environments.html) (CE). They are:  1.        Spot Capacity Optimized  2.        Best Fit Progressive  3.        Best Fit | |

|  |
| --- |
| **geoquery: a bridge between the gene expression omnibus (GEO) and bioconductor**  Bioinformatics, **14**, 1846–1847. <https://www.bioconductor.org/packages/release/bioc/html/GEOquery.html>  <https://www.bioconductor.org/packages/release/bioc/vignettes/GEOquery/inst/doc/GEOquery.html#platforms> |

**GOOD BIG PICTURE of Bioinformatics Software Tools**

|  |
| --- |
| **Collection of Bioinformatics Tools**  <https://github.com/jdidion/biotools/blob/main/README.md>  **Excellent** |
| Click [Raw] for actual Markdown text which renders as a good hyperlinked page |

**Mastering RNA-Seq (NGS Data Analysis) - A Critical Approach to Transcriptomic Data Analysis**

<https://www.youtube.com/watch?v=j4qpJ8sVjT0>

Video 03:18:55

Apr 17, 2018

This RNA-seq workshop will address critical issues related to Transcriptomics data, like:

Processing raw Next Generation Sequencing (NGS) data

|  |
| --- |
| **Relevant Journal Publications:**  [**https://journals.plos.org/ploscompbiol/search**](https://journals.plos.org/ploscompbiol/search)  **All Fields: bioconductor**  **Publication Date: 2016-01-01 - 2020-10-09**  **Sort By: Most Bookmarked**  **Subject Area: Principal component analysis**  >> results  **Ten quick tips for effective dimensionality reduction**  20 Jun 2019 PLOS Computational Biology  <https://doi.org/10.1371/journal.pcbi.1006907>  <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006907>  Citations: 19  Context Specific and Differential Gene Co-expression Networks via Bayesian Biclustering  28 Jul 2016 PLOS Computational Biology  <https://doi.org/10.1371/journal.pcbi.1004791>  Citations: 22  Machine learning-based microarray analyses indicate low-expression genes might collectively influence PAH disease  12 Aug 2019 PLOS Computational Biology  <https://doi.org/10.1371/journal.pcbi.1007264>  Citations: 2 |

NGS RNA-Seq bioinformatics tools, workflows

"differential expression"  
"exploratory data analysis"

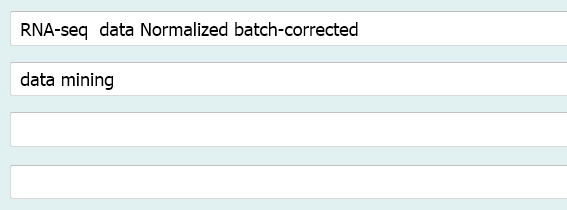
Attracted to the aspects of workflows following the pre-processing, normalization (Quantification of Mapped Reads, Differential Gene Expression).

The Application of Inferential Statistics, Machine Learning (PCA, Feature/Dimensional Reduction, Clustering).

**Google Search**

[**enabling cross study analysis of rna-sequencing data**](https://www.google.com/searchlr=&as_qdr=all&sxsrf=ALeKk00QSjfjp_4g_LC4EyNExgXHpC1w4w:1606327451528&q=enabling+cross+study+analysis+of+rna-sequencing+data&sa=X&ved=2ahUKEwi-jZCJpJ7tAhVkw1kKHXaKCWMQ1QIoAHoECAUQAQ)

<https://www.google.com/searchlr=&as_qdr=all&sxsrf=ALeKk00QSjfjp_4g_LC4EyNExgXHpC1w4w:1606327451528&q=enabling+cross+study+analysis+of+rna-sequencing+data&sa=X&ved=2ahUKEwi-jZCJpJ7tAhVkw1kKHXaKCWMQ1QIoAHoECAUQAQ>

  
<https://www.google.com/search?as_q=RNA-seq++data+Normalized+batch-corrected+&as_epq=data+mining+&as_oq=&as_eq=&as_nlo=&as_nhi=&lr=&cr=&as_qdr=all&as_sitesearch=&as_occt=any&safe=images&as_filetype=&tbs=>

**Keywords:**

RNA-seq

data mining

cross study analysis

data normalization

batch-corrected RNA-seq

**RNA-seq workflow: gene-level exploratory analysis and differential expression**

<http://master.bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html>

|  |
| --- |
|  |
| **Relevant journal publications**   |  | | --- | | **RNA-seq workflow: gene-level exploratory analysis and differential expression**  2015 Oct 14  <http://master.bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html> | | Abstract  Here we walk through an end-to-end gene-level RNA-Seq differential expression workflow using Bioconductor packages. We will start from the FASTQ files, show how these were aligned to the reference genome, and prepare a count matrix which tallies the number of RNA-seq reads/fragments within each gene for each sample. We will perform exploratory data analysis (EDA) for quality assessment and to explore the relationship between samples, perform differential gene expression analysis, and visually explore the results.  Keywords  RNA-seq, differential expression, gene expression, Bioconductor, statistical analysis, high-throughput sequencing, visualization, genomics | |  |  |  | | --- | | **RNA-seq: Basic Bioinformatics Analysis**  Curr Protoc Mol Biol. 2018 Oct  <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6168365/> | | The workflow includes three parts:  (a) mapping sequencing reads to a reference genome or transcriptome;  (b) quantifying expression levels of individual genes and transcripts;  (c) identifying specific genes and transcripts that are differentially expressed between samples.  Alignment/Mapping:  the sequence of each read to a reference genome, annotation of genes  - **STAR** alignment tool  Quality Assessment:  After mapping reads to the genome, it is important to survey the quality of the RNA-seq data  - **Picard**  - **SAMTools**  Quantification of Mapped Reads:  - **HTseq** to quantify sequencing reads mapped to each gene  (a) identify genes that are differentially expressed between conditions (sample groups),  (b) derive gene expression values for each individual transcript  Approaches for normalization include  CPM (counts per million reads),  RPKM (reads per kilobase per million reads),  FPKM (fragments per kilobase per million reads),  TPM (transcripts per million reads).  Differential Gene Expression Analysis:  **EdgeR** (Robinson et al., 2010)  **DESeq2** (Love et al. 2014)  Download and install required tools:  STAR: <https://github.com/alexdobin/STAR>  Picard: <https://broadinstitute.github.io/picard/>  HTseq: <https://htseq.readthedocs.io/en/release_0.9.1/install.html>  R: <https://www.r-project.org> |  |  | | --- | | **ngs.plot: Quick mining and visualization of next-generation sequencing data by integrating genomic databases**  BMC Genomics 2014 volume 15, Article number: 284  415 Citations  <https://bmcgenomics.biomedcentral.com/articles/10.1186/1471-2164-15-284> | |  |  |  | | --- | | **pcaExplorer: an R/Bioconductor package for interacting with RNA-seq principal components**  BMC Bioinformatics 2019 volume 20, Article number: 331  17 Citations  <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2879-1> | | **pcaExplorer**  <http://bioconductor.org/packages/release/bioc/html/pcaExplorer.html>  **pcaExplorer User Guide**  27 October 2020  <http://bioconductor.org/packages/release/bioc/vignettes/pcaExplorer/inst/doc/pcaExplorer.html>  **Up and running with pcaExplorer**  27 October 2020  <http://bioconductor.org/packages/release/bioc/vignettes/pcaExplorer/inst/doc/upandrunning.html>  data component (count matrix, experimental data, dds object, annotation) |      |  | | --- | | **A Beginner’s Guide to Analysis of RNA Sequencing Data**  <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6096346/>  Am J Respir Cell Mol Biol. 2018 Aug | | **Abstract:**  A general understanding of the principles underlying each step of RNA-seq data analysis allows investigators without a background in programming and bioinformatics to critically analyze their own datasets as well as published data. Our goals in the present review are to break down the steps of a typical RNA-seq analysis and to highlight the pitfalls and checkpoints along the way that are vital for bench scientists and biomedical researchers performing experiments that use RNA-seq.  **Keywords:**  RNA sequencing, transcriptomics, bioinformatics, data analysis  **Associated Data:**  The RNA-seq data reported in this article has been deposited in NCBI’s Gene Expression Omnibus (GEO) and are accessible through GEO Series accession number GSE116583.  **Experimental Design and Approach:**  A major goal of RNA-seq analysis is to identify differentially expressed and coregulated genes and to infer biological meaning for further studies. Source material can be cells cultured in vitro, whole-tissue homogenates, or sorted cells. The ability to interpret findings depends on appropriate experimental design, implementation of controls, and correct analysis. Every effort should be made to minimize batch effect, because small and uncontrolled changes in an environment can result in identification of differentially expressed genes (DEGs) unrelated to the designed experiment. Sources of batch effect can occur during the experiment, during the RNA library preparation, or during the sequencing run and include but are not limited to those listed in Table 1. Once a well-designed and controlled experiment is performed, a structured approach to the dataset allows for quality control followed by unbiased analysis of the data. In the present analysis, we use an approach that includes setting low count filtering, establishing a noise threshold, checking for potential outliers, running appropriate statistical tests to identify DEGs, clustering of genes by expression pattern, and testing for gene ontology (GO) enrichment. For each of these analysis components, we aim to highlight important checkpoints and quality controls that will streamline and strengthen data analysis, avoid bias, and allow investigators to maximally use their datasets.  **Clustering**  The two most common clustering methods used for RNA-seq data analysis are hierarchical and k-means clustering (see Clustering box). The most common form of hierarchical clustering is a bottom-up agglomerative approach that organizes the data into a tree structure without user input by starting with each data point as its own cluster and iteratively combining them into larger clusters or “clades.” In contrast, k-means clustering requires the investigator to define the number of clusters (k) a priori, and data are then sorted into the cluster with the nearest mean. It is possible to assess a range of k-values to decide how to best capture the trends. In addition, various tools such as Silhouette exist to help the investigator determine the ideal k-value, but some subjectivity remains (21). By adjusting the k, the investigator may set the degree of granularity they would like to achieve with the data. For either approach, the user must specify the distance metric by which data points are considered similar. Typically, Pearson’s correlation is used, and this is generally the default in software designed for RNA-seq analysis. Both approaches are widely used, and both aid the investigator in identifying groups of genes that display similar expression patterns, allowing for further downstream analyses. The clusters can then be used as input for an analysis of functional enrichment (see next section).  Why do we use clustering on RNA-seq data?  Clustering of RNA-seq data may be used to identify patterns of gene expression by grouping genes based on their distance in an unsupervised manner. Clustering RNA-seq data is used as an exploratory tool that allows the user to organize and visualize relationships between groups of genes, and to select certain genes for further consideration.  **Hierarchical clustering**  The most commonly used hierarchical clustering approach is a form of agglomerative, or bottom-up, clustering that iteratively merges clusters (originally consisting of individual data points) into larger clusters or “clades”.  **K-means clustering**  Data points are iteratively partitioned into clusters based on the minimum distance to the cluster mean. The number of clusters (k) is set by the investigator. |      |  | | --- | | **Review of RNA-Seq Data Analysis Tools**  February 17, 2016  <https://rna-seqblog.com/review-of-rna-seq-data-analysis-tools/> |      |  | | --- | | **Feature-based classification of human transcription factors into hypothetical sub-classes related to regulatory function**  BMC Bioinformatics volume 17, Article number: 459 (2016)  7 Citations  <http://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1349-2> |  |  | | --- | | **GEO2RNAseq: An easy-to-use R pipeline for complete pre-processing of RNA-seq data**  September 16, 2019 bioRxiv Preprint  <https://www.biorxiv.org/content/10.1101/771063v1.full> |  |  | | --- | | **RNAseq data analysis in R - Notebook**  <http://monashbioinformaticsplatform.github.io/RNAseq-DE-analysis-with-R/RNAseq_DE_analysis_with_R.html> | | Install and load packages  Mapping reads to a reference genome  Count reads for each feature  QC and stats  Differential Expression  Gene Annotation  Gene Set Enrichment |  |  | | --- | | **Introduction to differential gene expression analysis using RNA-seq**  September 2015 updated November 14, 2019  <https://chagall.med.cornell.edu/RNASEQcourse/Intro2RNAseq.pdf> | |

|  |
| --- |
|  |
| |  | | --- | | **Google**  **enabling cross-study analysis of RNA-Sequencing data**  [https://www.google.com/search?lr=&as\_qdr=all&sxsrf=AleKk00QSjfjp\_4g\_LC4EyNExgXHpC1w4w:1606327451528&q=enabling+cross+study+analysis+of+rna-sequencing+data&sa=X&ved=2ahUKEwi-jZCJpJ7tAhVkw1kKHXaKCWMQ1QIoAHoECAUQAQ](https://www.google.com/search?lr=&as_qdr=all&sxsrf=ALeKk00QSjfjp_4g_LC4EyNExgXHpC1w4w:1606327451528&q=enabling+cross+study+analysis+of+rna-sequencing+data&sa=X&ved=2ahUKEwi-jZCJpJ7tAhVkw1kKHXaKCWMQ1QIoAHoECAUQAQ)  **Google**  **RNA-Seq data mining** | | **A survey of best practices for RNA-seq data analysis**  Genome Biology volume 17, Article number: 13 (2016)  769 Citations  <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0881-8>  **Massive mining of publicly available RNA-seq data from human and mouse**  Nature Communications volume 9, Article number: 1366 (2018)  96 Citations  <https://www.nature.com/articles/s41467-018-03751-6>  **Omics Playground: Explore Omics Data Freely**  <https://omicsplayground.readthedocs.io/en/latest/index.html>  Omics Playground is a comprehensive self-service platform platform for visualization, analytics and exploration of Big Omics Data. It allows users to apply a multitude of state-of-the-art analysis tools to their own data to explore and discover underlying biology in a short time.  The platform offers a unique combination of features that distinguishes it from the other analytics platforms currently available. We believe that data preprocessing (primary analysis) and statistical testing (secondary analysis) are now well established, and the most challenging task is currently data interpretation (tertiary analysis) that often takes the longest time but where actual insights can be gained. Therefore, Omics Playground focuses strongly on tertiary analysis while providing good support for secondary analysis.  Reanalyzing Public Datasets  To illustrate the use case of the Omics Playground, we reanalyzed different types of publics datasets, including microarray, bulk RNA-seq, single-cell RNA-seq and proteomic datasets to recapitulate the results.  <https://omicsplayground.readthedocs.io/en/latest/examples/examples.html> |  |  | | --- | | **comparing rna-seq datasets** | | **Broad Institute Gene Set Enrichment Analysis (GSEA)**  <https://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Using_RNA-seq_Datasets_with_GSEA>  **BioStars**  **Question: RNA-seq data comparison across experiments**  <https://www.biostars.org/p/117451/>  **BioStars**  **Question: Comparing similarity of RNAseq datasets**  <https://www.biostars.org/p/334235/>  **Analysis of public RNA-sequencing data reveals biological consequences of genetic heterogeneity in cell line populations**  Scientific Reports volume 8, Article number: 11226 (2018) Cite this article  7 Citations  <https://www.nature.com/articles/s41598-018-29506-3> |   **Books:**  **NCBI Books Advanced Search**  <https://www.ncbi.nlm.nih.gov/books/advanced/>   |  | | --- | | **Computational Biology**  Brisbane (AU): Codon Publications; 2019 Nov 21.  ISBN-13: 978-0-9944381-9-5  <https://www.ncbi.nlm.nih.gov/books/NBK550339/>  Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)  <https://creativecommons.org/licenses/by-nc/4.0/> | | Chapter 4 Biological Sequence Analysis  <https://www.ncbi.nlm.nih.gov/books/NBK550342/>  Chapter 5 Multivariate Statistical Methods for High-Dimensional Multiset Omics Data Analysis  <https://www.ncbi.nlm.nih.gov/books/NBK550343/>  Chapter 6 Statistical Methods for RNA Sequencing Data Analysis  <https://www.ncbi.nlm.nih.gov/books/NBK550334/> | |

**RNA-seq workflow: gene-level exploratory analysis and differential expression**

<http://master.bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html>

|  |
| --- |
| **NCBI Reference Sequence Database (RefSeq)**  <https://www.ncbi.nlm.nih.gov/refseq/>  <https://ftp.ncbi.nlm.nih.gov/genomes/refseq/vertebrate_mammalian/Homo_sapiens/> |
|  |

|  |
| --- |
|  |
| **Most Common Tools Used for the Analysis of WGS Data**  <https://www.researchgate.net/figure/NGS-and-analysis-pipelines-Most-common-tools-used-for-the-analysis-of-WGS-data-QC_fig2_317413533> **Comprehensive Outline of Whole Exome Sequencing Data Analysis Tools** **Available in Clinical Oncology** <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6895801/>  **Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data** 2019  <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2599-6>  **Most Common Tools Used for the Analysis of WGS Data**  <https://www.researchgate.net/figure/NGS-and-analysis-pipelines-Most-common-tools-used-for-the-analysis-of-WGS-data-QC_fig2_317413533> **Comprehensive Outline of Whole Exome Sequencing Data Analysis Tools Available in Clinical Oncology** <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6895801/>  **Comparitive Analysis of Differential Gene Expression Analysis Tools for Single-Cell Sequencing Data**  <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2599-6>  **FASTQC A quality control tool for high throughput sequence data**. 2014 September 29  <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>  **Free RNA-seq Analysis Tools – Software and Resources**  <https://bioinformaticshome.com/tools/rna-seq/rna-seq.html> |

|  |
| --- |
|  |
| **Most Common Tools Used for the Analysis of WGS Data**  <https://www.researchgate.net/figure/NGS-and-analysis-pipelines-Most-common-tools-used-for-the-analysis-of-WGS-data-QC_fig2_317413533> **Comprehensive Outline of Whole Exome Sequencing Data Analysis Tools** **Available in Clinical Oncology** <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6895801/>  **Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data** 2019  <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2599-6>  **Most Common Tools Used for the Analysis of WGS Data**  <https://www.researchgate.net/figure/NGS-and-analysis-pipelines-Most-common-tools-used-for-the-analysis-of-WGS-data-QC_fig2_317413533> **Comprehensive Outline of Whole Exome Sequencing Data Analysis Tools Available in Clinical Oncology** <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6895801/>  **Comparitive Analysis of Differential Gene Expression Analysis Tools for Single-Cell Sequencing Data**  <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2599-6>  **FASTQC A quality control tool for high throughput sequence data**. 2014 September 29  <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>  **Free RNA-seq Analysis Tools – Software and Resources**  <https://bioinformaticshome.com/tools/rna-seq/rna-seq.html> |

|  |
| --- |
| **Analysis of single cell RNA-seq data (**University of Cambridge Bioinformatics)  <https://biocellgen-public.svi.edu.au/mig_2019_scrnaseq-workshop/public/index.html>  quality control, visualisation, data normalisation, exploratory data analysis, clustering, trajectory (pseudotime) inference, differential expression, batch correction, combining datasets, data integration, confounders, latent spaces, cell annotation, case studies |
| **2-day Course: (16 hours of video) RStudio**  Day 1: <https://www.youtube.com/watch?v=thHgPqQpkE4&feature=emb_err_woyt>  Processing Raw scRNA-Seq Data  Construction of Expression Matrix  Intro to R/Bioconductor  Seurat  Day 2: <https://www.youtube.com/watch?v=7dQ_pleDO2Y&feature=emb_err_woyt>  Clustering example  Feature Selection  Pseudotime Analysis  Differental Expression Analysis  DE Real Dataset  Comparing/Combining scRNA  Search scRNA-Seq Data  Seurat  scRNA-Seq Pipeline |

|  |
| --- |
| **Principal Components Analysis / Feature Selection**  **bioRxiv**  <https://www.biorxiv.org/>  <https://www.biorxiv.org/search> |
| **mixOmics: an R package for omics feature selection and multiple data integration Aug 2017**  <https://www.biorxiv.org/content/10.1101/108597v4.full>  **Differential Principal Components Reveal Patterns of Differentiation in Case/Control Studies Feb 2019**  <https://www.biorxiv.org/content/10.1101/545798v1.full>  **pathwayPCA: an R package for integrative pathway analysis with modern PCA methodology and gene selection April 2019**  <https://www.biorxiv.org/content/10.1101/615435v1.full>  **Accurate and Fast feature selection workflow for high-dimensional omics data June 2017**  <https://www.biorxiv.org/content/10.1101/144162v1.full> |

|  |
| --- |
| **SEURAT: R toolkit for single cell genomics**  <https://satijalab.org/seurat/v3.1/pbmc3k_tutorial.html> |
| Installation Instructions for Seurat  <https://satijalab.org/seurat/install.html>  Vignettes: Guided Analyses  <https://satijalab.org/seurat/vignettes.html>  **Seurat - Guided Clustering Tutorial** 2020 April  Setup the Seurat Object  Standard pre-processing workflow  Normalizing the data  Identification of highly variable features (feature selection)  Scaling the data  Perform linear dimensional reduction  Determine the ‘dimensionality’ of the dataset  Cluster the cells  Run non-linear dimensional reduction (UMAP/tSNE)  Finding differentially expressed features (cluster biomarkers)  Assigning cell type identity to clusters |

|  |
| --- |
| **Statistics for Genomics**  <https://www.youtube.com/playlist?list=PLdl4u5ZRDMQQpUcSDRKN3V2vvx3_SmMbr> |
| **17 Videos:** 2017 May  **Statistics for Genomcs: Distances and Clustering**  <https://www.youtube.com/watch?v=wQhVWUcXM0A&list=PLdl4u5ZRDMQQpUcSDRKN3V2vvx3_SmMbr&index=2>  **Statistics for Genomics Lab: Quick Introduction to R and Bioconductor**  <https://www.youtube.com/watch?v=J5h5WxOn3Gw&list=PLdl4u5ZRDMQQpUcSDRKN3V2vvx3_SmMbr&index=11>  **Statistics for Genomics Lab: Distances and Clustering RStudio**  <https://www.youtube.com/watch?v=PArRvqLUP6o&list=PLdl4u5ZRDMQQpUcSDRKN3V2vvx3_SmMbr&index=7>  **Statistics for Genomics: Introduction to RNAseq**  <https://www.youtube.com/watch?v=C8RNvWu7pAw&list=PLdl4u5ZRDMQQpUcSDRKN3V2vvx3_SmMbr&index=12>  **Statistics for Genomics: Advanced Differential Expression**  <https://www.youtube.com/watch?v=QINX3cI7qgk&list=PLdl4u5ZRDMQQpUcSDRKN3V2vvx3_SmMbr&index=15>  **Statistics for Genomics: Useful plots and bad plots**  <https://www.youtube.com/watch?v=46-t2jOYsyY&list=PLdl4u5ZRDMQQpUcSDRKN3V2vvx3_SmMbr&index=17> |

|  |
| --- |
| **BioC 2020 Conference**  <http://bioc2020.bioconductor.org/schedule> |
| 100: **Annotating inter-sample DNA methylation and ATAC-seq variation with COCOA**  100: **Human RNA-seq data from recount2 and related packages**  100: **Introduction to Bioconductor annotation resources**  100: **A tidy transcriptomics introduction to RNA-Seq analyses**  200: **Functional enrichment analysis of high-throughput omics data**  200: **Best practices for ATAC-seq QC and data analysis**  200: **Copy number variation analysis with Bioconductor**  200: **Interactive visualization of SummarizedExperiment objects with iSEE**  200: **Integrated ChIP-seq data analysis workshop**  200: **An introduction to matrix factorization and principal component analysis in R**  500: **Bioconductor toolchain for usage and development of reproducible bioinformatics pipelines in CWL**  500: **Effectively Using the DelayedArray Framework to Support the Analysis of Large Datasets**  100: **Cloud-based genomics using Terra/AnVIL** |

|  |
| --- |
| **Bioconductor: DESeq2** Differential gene expression analysis based on the negative binomial distribution |
| **Analyzing RNA-seq data with DESeq2** 2020 Oct  <https://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html>  **RNA-seq workflow: gene-level exploratory analysis and differential expression** 2019 Oct  <http://master.bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html>  **Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2** 2014 Dec  <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-014-0550-8>  **The R package Rsubread is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads**  <https://academic.oup.com/nar/article/47/8/e47/5345150>  **Data preprocessing and creation of the data objects pasillaGenes and pasillaExons** 2020 May  <http://bioconductor.org/packages/release/data/experiment/vignettes/pasilla/inst/doc/create_objects.html> |
|  |

### Reproducible and robust workshop materials

Workshops were authored using R Markdown, and compiled into a book (PDF and ePub) and website using Bookdown R package. Bookdown, in turn, uses the gitbook publishing system ( <https://www.gitbook.com/>) to produce a variety of formats from the same source material. R Markdown files intended to be part of a Bookdown project do not contain the required front matter of a typical stand-alone R Markdown document. To help authors use and test the correct format, we seeded each workshop document with the syllabus that had been submitted by that author, and successfully built the book of the submitted syllabi. Each workshop represented a chapter of a book compiled using the Bookdown software. This approach provided several advantages:

* R markdown syntax is already familiar to any developer of a Bioconductor package, since it is the standard approach to creating the package “vignette” or prose documentation.
* R markdown implements “literate programming” by including formatted text, runnable code, and output of the code
* Bookdown allows collating chapters as a clean, lightweight online book format, and pandoc additionally allows creation of PDF and ePub formats
* These formats can then be self-published with options to order paper copies through companies such as <https://leanpub.com>

This approach allowed automatic installation of required packages by listing them in the DESCRIPTION file required by R packages.

**bookdown: Authoring Books and Technical Documents with R Markdown**

<https://books.google.com/books?hl=en&lr=&id=_LrZDQAAQBAJ&oi=fnd&pg=PT12&dq=+Xie+Y+:+bookdown:+Authoring+Books+and+Technical+Documents+with+R+Markdown+%5BInternet%5D+.+Boca+Raton,+Florida:+Chapman+and+Hall/CRC%3B+2016++10.1201/9781315204963++&ots=tz1AhTAN96&sig=mmTuwDYYrl3sVm8SjtJD8tvnwd0#v=onepage&q&f=false>