Leveraging Cloud-based and Machine Learning Tools for Genomic Data Science Research

Babu Guda, Ph.D.

Director, GP IDeA-CTR BIBCE Core
Assistant Dean for Research Development
Professor & Vice-Chair, Genetics, Cell Biology & Anatomy
College of Medicine
University of Nebraska Medical Center

Finding the needles in a haystack

- Human body – Systems – Organs –Tissues –Cells: ~37 trillion cells ($37x10^{12}$)
- Cells – about 100 different types of cells with different functions
- Each cell –Nucleus –Chromosomes –Genes –DNA – 6 billion letters of DNA (ATGC)
- Mutations cause diseases – Heterogeneous diseases like cancer, we need to find mutations in millions of cells, where each cell has billions of DNA letters

Broader Areas of Research Expertise

- Novel algorithm and tool development
- Cancer genomics data analysis
- Virtual screening and cheminformatics
- Multiome data analysis (Exome, transcriptome, methylation, proteome, metabolome, metagenome, and single-cell RNA-seq)
- Precision medicine (genomics-based)
- Machine-learning (ML), deep-learning (DL) and AI applications
- Database, web application, and software development
- High-performance computing (CPU and GPU computing)
- Cloud computing and application development

New tools and software developed since 1999

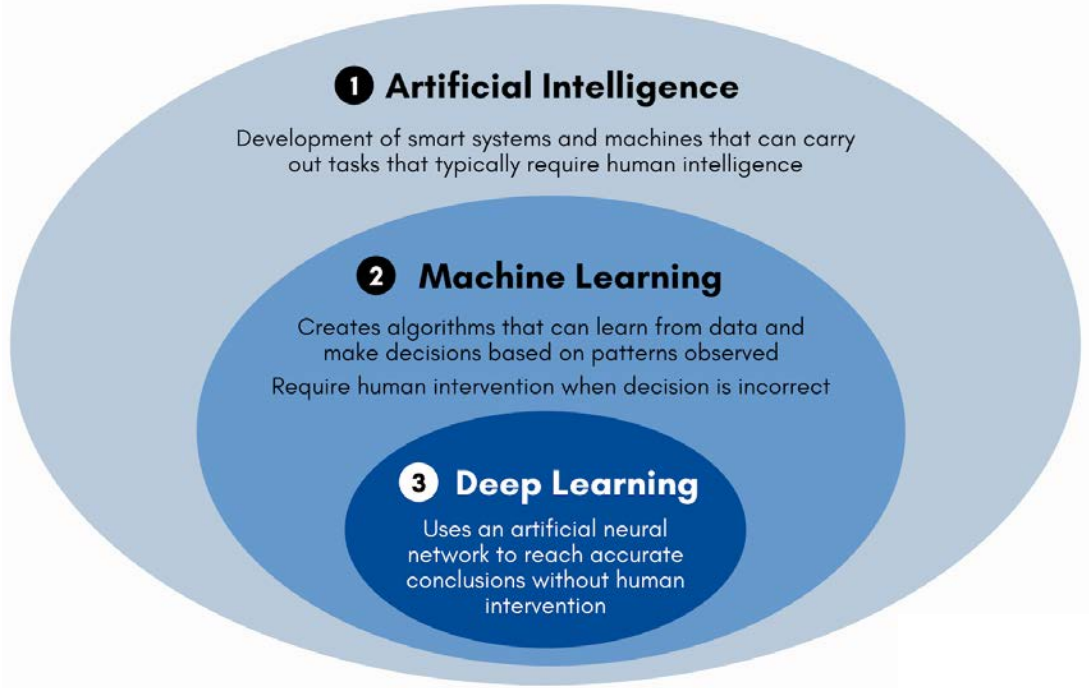
Project Period	Name of the Tool	Description
2022-23	iMDGAVis	A shiny web application to efficiently perform popular visualization and analysis for Metagenomic data
2022-23	scATAC-seq	A cloud-based learning module to analyze ATAC-seq and single cell ATAC-seq data
2019-22	mintRULS	Prediction of miRNA-mRNA Target Site Interactions Using Regularized Least Square Method
2018-22	NBBt-test	A versatile method for differential analysis of multiple types of RNA-seq data
2016-21	StrainIQ	Taxonomic identification of quantification of microbiome sequencing data
2014-20	NNTC-DCC	National NeuroAIDS Tissue Consortium Data Coordinating Center
2014-20	RedPanda	A novel method for variant calling using single cell RNA-sequencing data
2012-17	ChimeRScope	A novel tool for predicting fusion genes using gene fingerprints
2012-15	LocSigDB	A comprehensive database of protein localization signals
2011-15	ECemble	An enzyme classification algorithm and software tool
2011-13	MetaID	A method and tool for taxonomic profiling of metagenomic data
2012-13	Nebraska BioBank	Database and software to de-identify patient records and link blood samples to their Electronic Medical Record Data
2011-14	Cancer PPIs	Cancer protein interaction networks analysis using data mining
2007-09	DDI prediction	A method for predicting domain-domain interactions in proteins
2006-12	ngLOC	A novel method with validation, software and web server for predicting protein subcellular localization from sequence data
2005-06	DMAPS	A database of multiple alignments for protein structures
2004-06	pTARGET	A new method and web server for protein localization prediction
2003-04	SledgeHMMER	A web server for batch searching of Pfam database
2003-05	MITOPRED	A new method and web server for predicting mitochondrial proteins
1999-04	CE-MC	A novel method, software and web server for multiple protein structure alignment

Machine Learning vs. AI

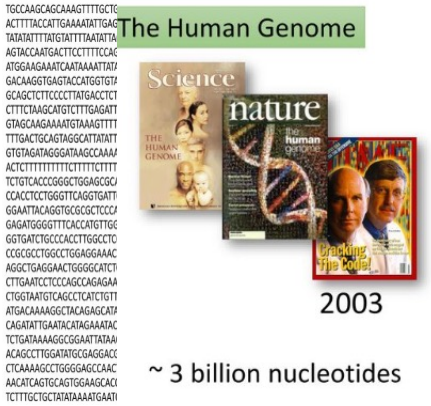
Artificial Intelligence (AI) refers to *autonomous systems capable of performing tasks* that otherwise require human intelligence and judgement (includes ML, neural networks, robotics, expert systems, NLP, etc.).

Machine Learning (ML) is a branch of AI and computer science which focuses on the *use of data and algorithms to imitate the way that humans learn*, gradually improving its accuracy.

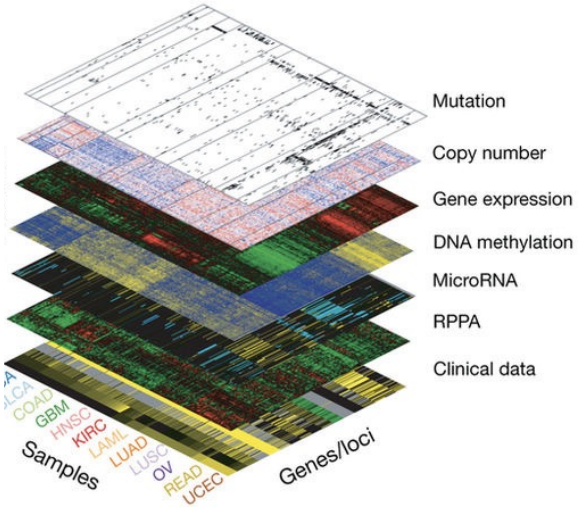
Deep Learning (DL) learns by discovering intricate structures in the data. DL employs multiple processing layers to create *multiple levels of abstraction to represent the data*.

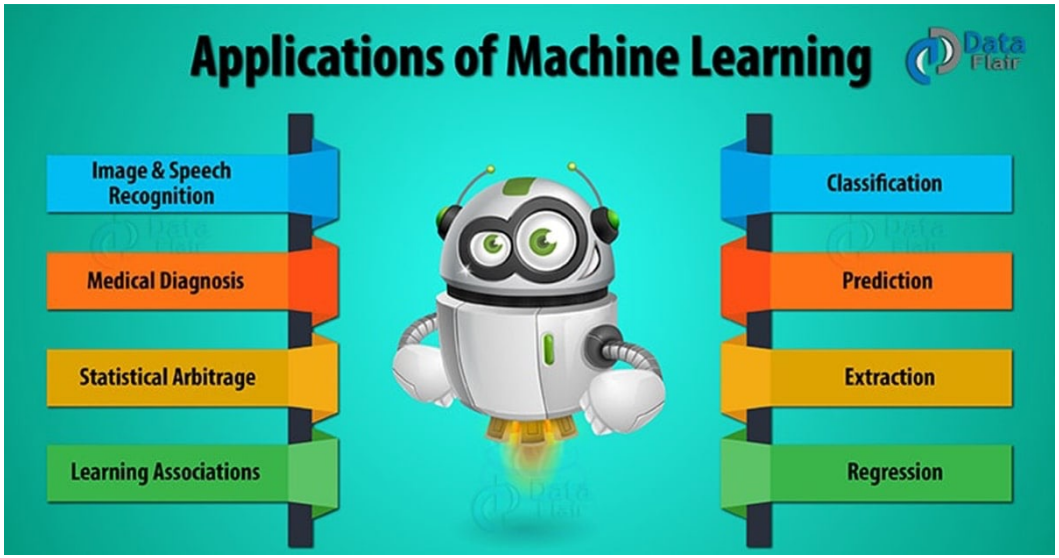


Genomic era and Machine Learning



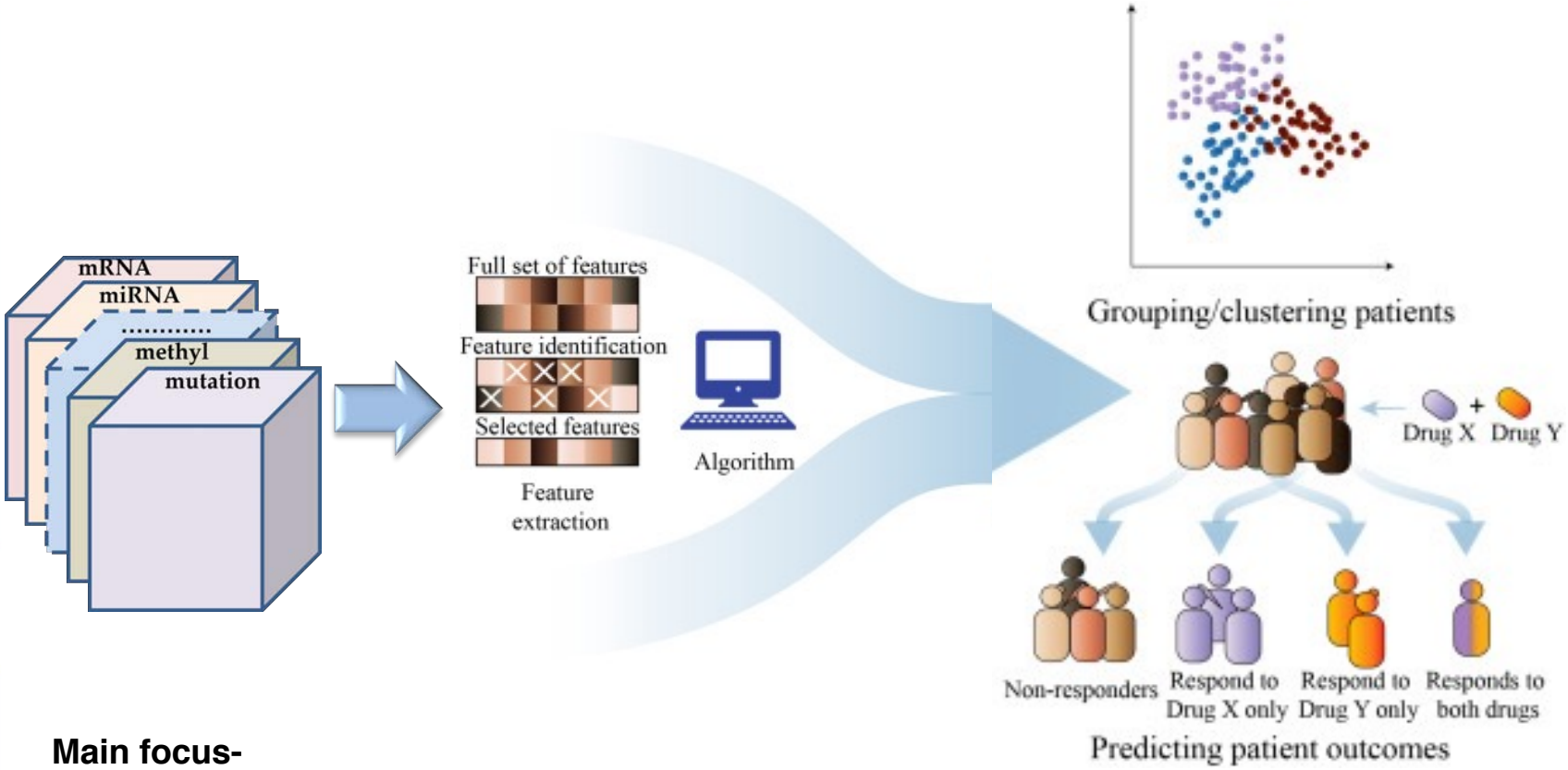
Multiomics





https://data-flair.training/blogs/machine-learning-applications/

My Research Interests Using Machine Learning: A 30,000-foot view

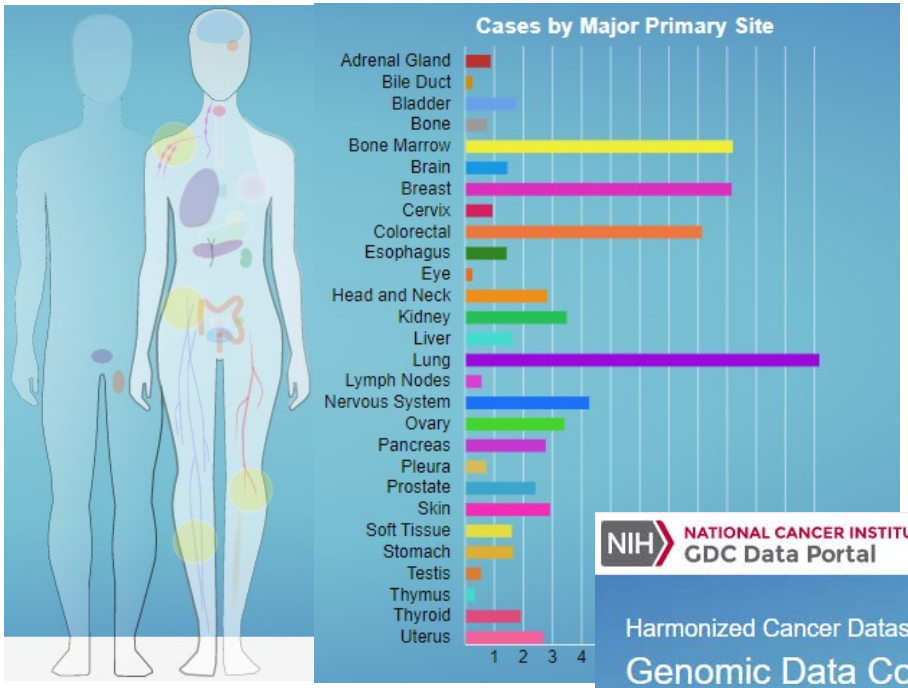


Main focus-

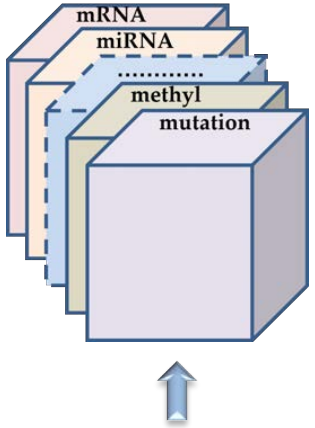
- Utilizing patients' multiomics data and machine learning methods to
  - correlate genotype/phenotype
  - transform genomics information into clinically actionable targets
- Developing novel algorithmic strategies to best utilize omics information

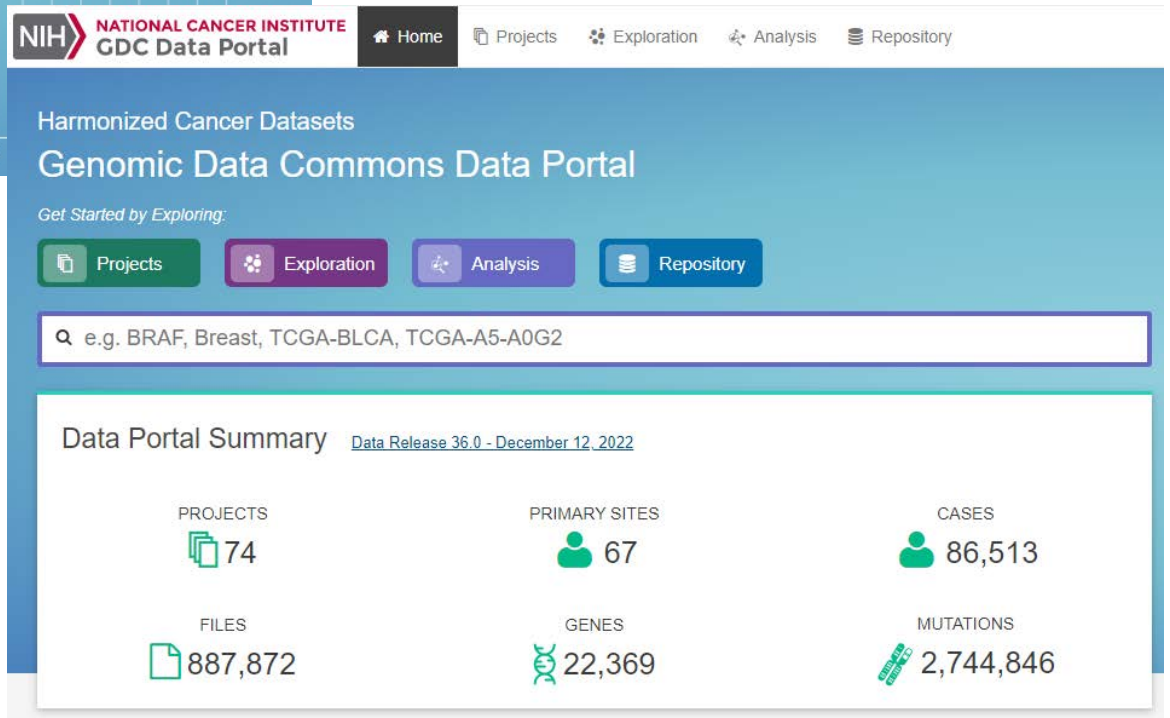
Figure partly adopted from Toh *et al.*, 2019

Data resources



Multiomics and clinical data





Other resources

cBioPortal FOR CANCER GENOMICS

GEO Gene Expression Omnibus

All of Us RESEARCH PROGRAM

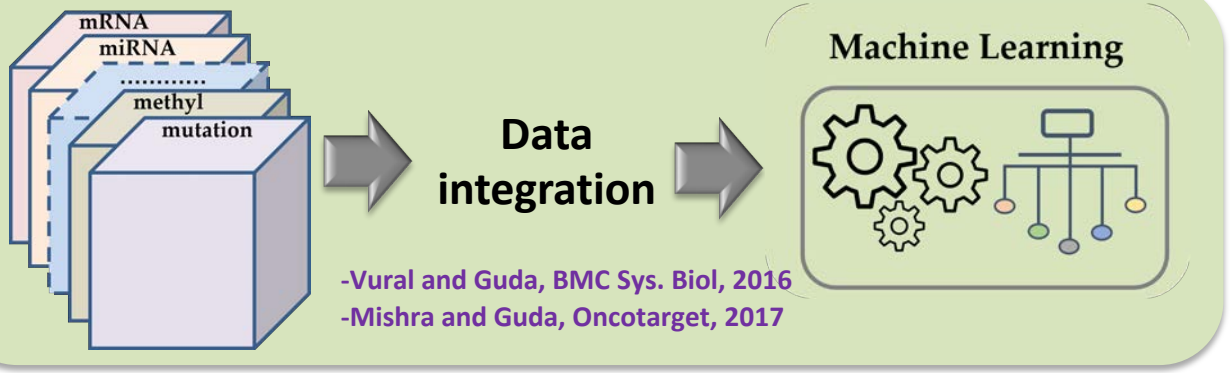
COSMIC Catalogue Of Somatic Mutations In Cancer

Machine Learning Projects

Guda Lab

King and Guda, Genome Biology, 2007
(supervised learning using Bayesian approach)

Brian King
(former PhD student)



-Vural and Guda, BMC Sys. Biol, 2016
-Mishra and Guda, Oncotarget, 2017

Unsupervised Learning

Major cancers
(subtyping, survival & clinical correlation)

Sushil Shakyawar
(Postdoc)
-Shakyawar et al., (in revision)

Osteosarcoma
(subtyping, biomarkers)

-Southekal et al., Cancers, 2023
Siddesh Southekal
(former PhD student)

Supervised Learning

Breast Cancer
(subtyping, biomarkers)

-Patel et al., (in submission)
Jai Chand Patel
(Postdoc)

COVID-19
(diagnosis, subtyping, biomarkers)

Sahil Sethi
(PhD student)

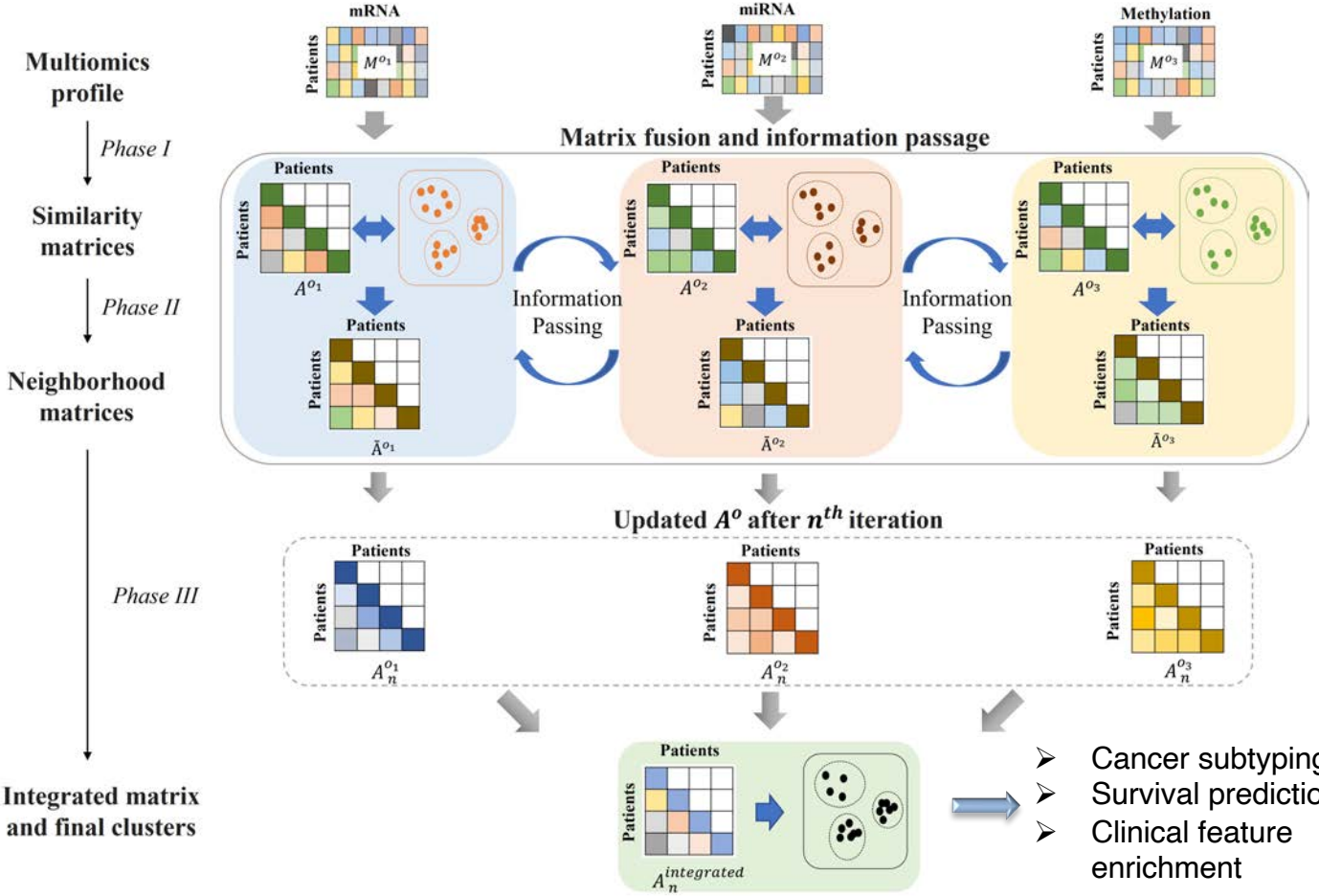
Iterative cluster fusion (iCluF): unsupervised clustering using multiomics data

Shakyawar et al., (in revision)

Sushil Shakyawar
(Postdoc)

Main objective

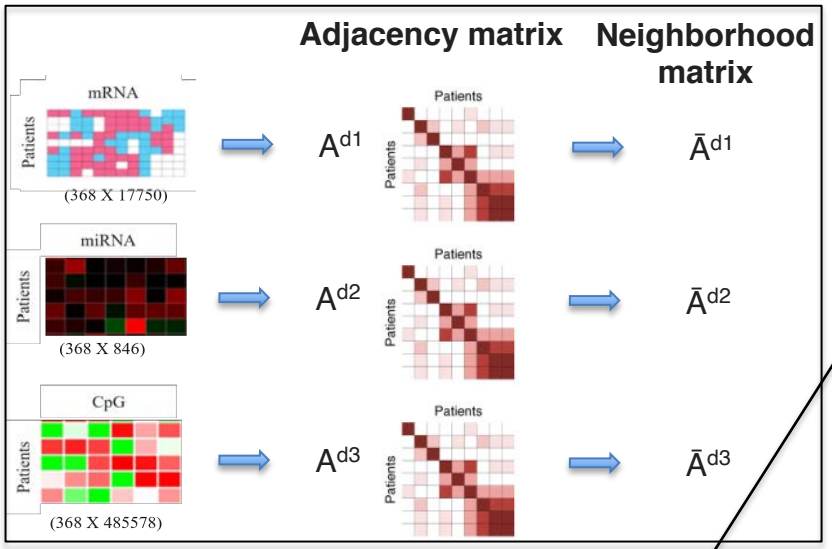
Integrate multiple data types and capture both shared and complementary information from each type of data.



- Cancer subtyping
- Survival prediction
- Clinical feature enrichment

iCluF (continued)

Integration strategy for merging neighborhood matrices



Status of Adjacency matrix A
(after n+1 iteration)

Iteration

Adding effect (from $data_2$) **Adding effect** (from $data_3$)

at n^{th} iteration

$$A_{n+1}^{d1} = \bar{A}_n^{d1}\, A_n^{d2}\, (\bar{A}_n^{d1})^T + \bar{A}_n^{d1}\, A_n^{d3}\, (\bar{A}_n^{d1})^T$$

Neighborhood matrix^T **($data_1$)**

Adj matrix **($data_2$)**

Neighborhood matrix **($data_1$)**

$$A_{n+1}^{d2} = \bar{A}_n^{d2}\, A_n^{d1}\, (\bar{A}_n^{d2})^T + \bar{A}_n^{d2}\, A_n^{d3}\, (\bar{A}_n^{d2})^T$$

$$A_{n+1}^{d3} = \bar{A}_n^{d3}\, A_n^{d1}\, (\bar{A}_n^{d3})^T + \bar{A}_n^{d3}\, A_n^{d2}\, (\bar{A}_n^{d3})^T$$

Integrated matrix $$A_{n+1}^{integrate} = \frac{(A_{n+1}^{d1} + A_{n+1}^{d2} + A_{n+1}^{d3})}{3}$$

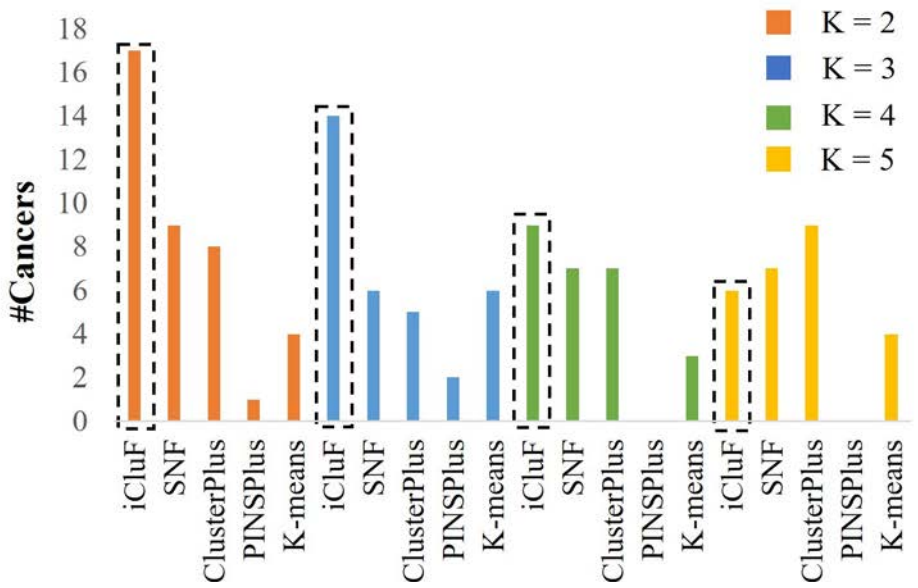
iCluF (continued)

Shakyawar et al., (submitted to Bioinformatics)

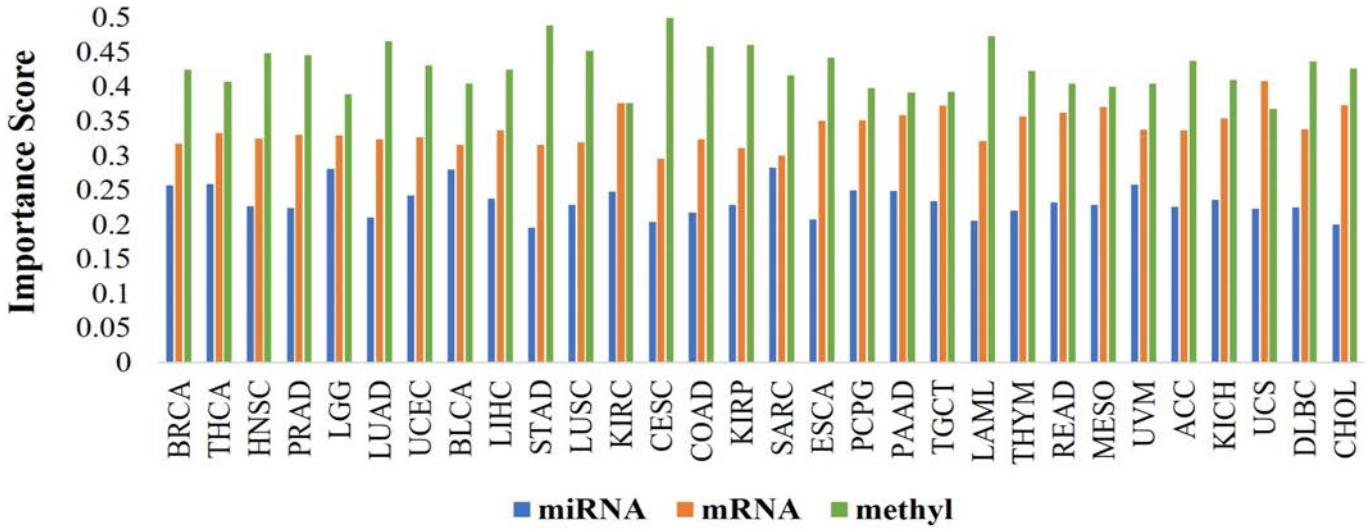
Comparison of iCluF with other methods

(Total # of cancer types tested = 30)

- SNF
- ClusterPlus
- PINSPlus
- K-means



Contribution of individual of 'Omics features to prediction of subtypes

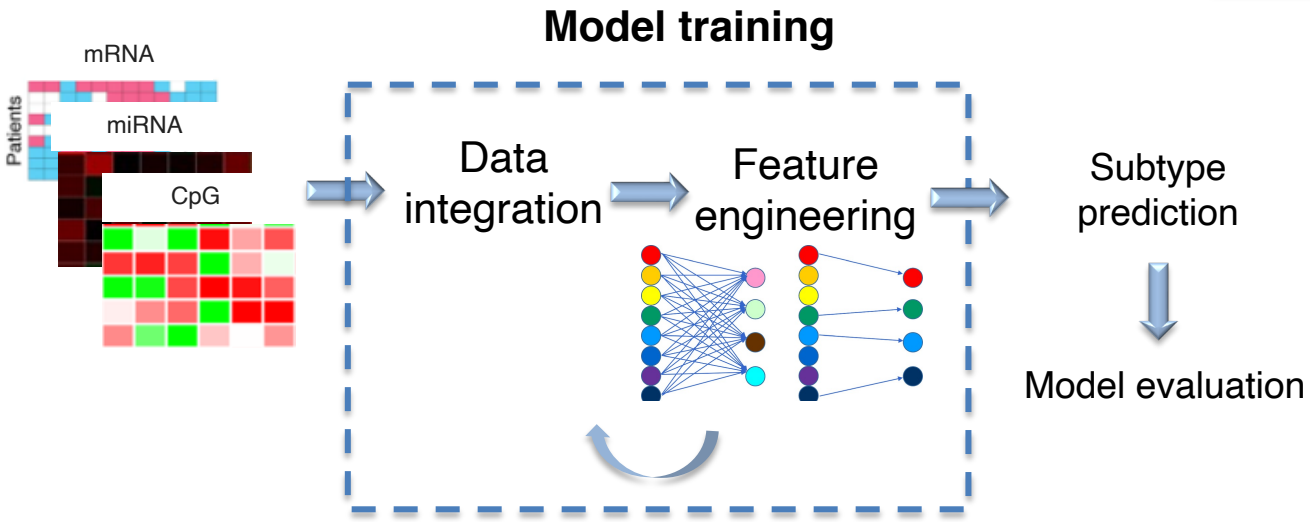


GAIN-BRCA: A Graphical Explainable AI-Net Framework for Breast Cancer Subtype classification

Jai Chand Patel
(Postdoc)

Main objective

Capture the hidden patterns in multiomics datasets to predict cancer subtypes using supervised learning



The algorithmic strategy is crucial to identify set of feature(s) for cancer subtype prediction

GAIN-BRCA (continued)

Table 1. The number of PAM50-based intrinsic subtypes of TCGA-BRCA.

BRCA Subtype	Number of Samples
Luminal A	393
Luminal B	124
HER2+	38
Basal	114
Total	**669**

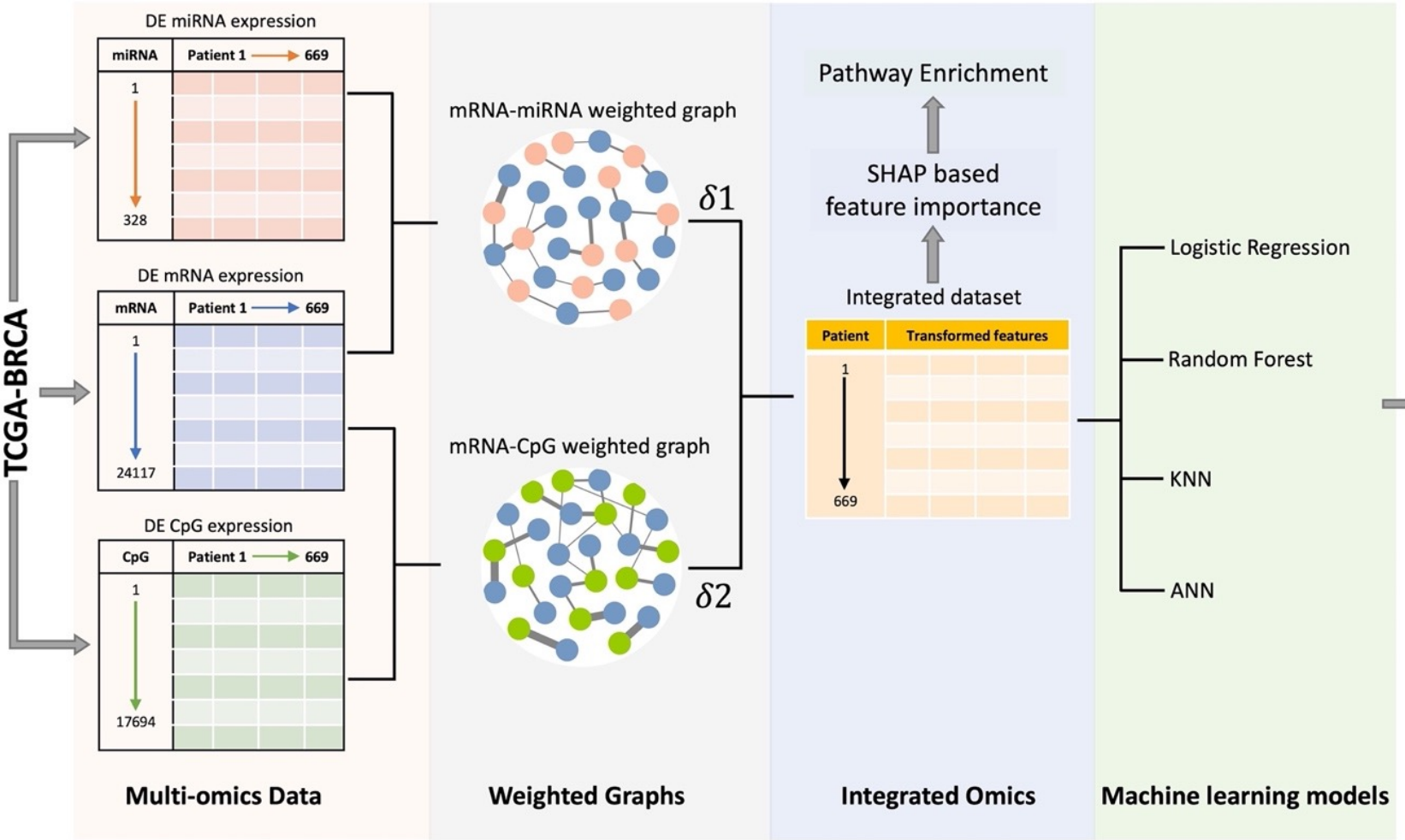
Table 2. The overview of omic-specific features of processed TCGA-BRCA dataset.

Omics type	# of features	Workflow
mRNA	24,117	STAR – Counts
miRNA	328	BCGSC miRNA Profiling
DNA Methylation	17,694	SeSAMe Methylation Beta Estimation

GAIN-BRCA (continued)

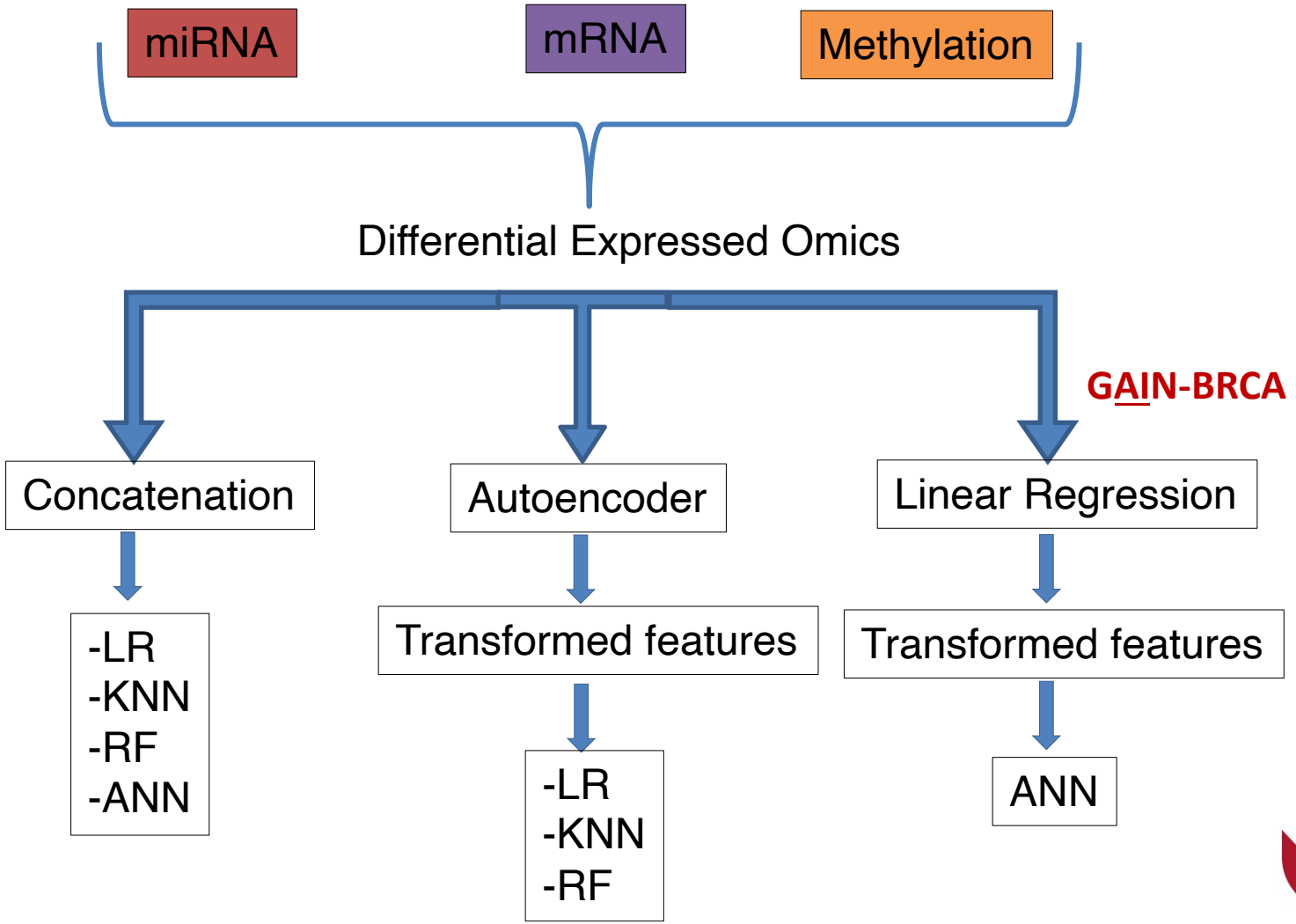
Patel et al., (To be sumitted)

Deep-Learning based algorithmic strategy

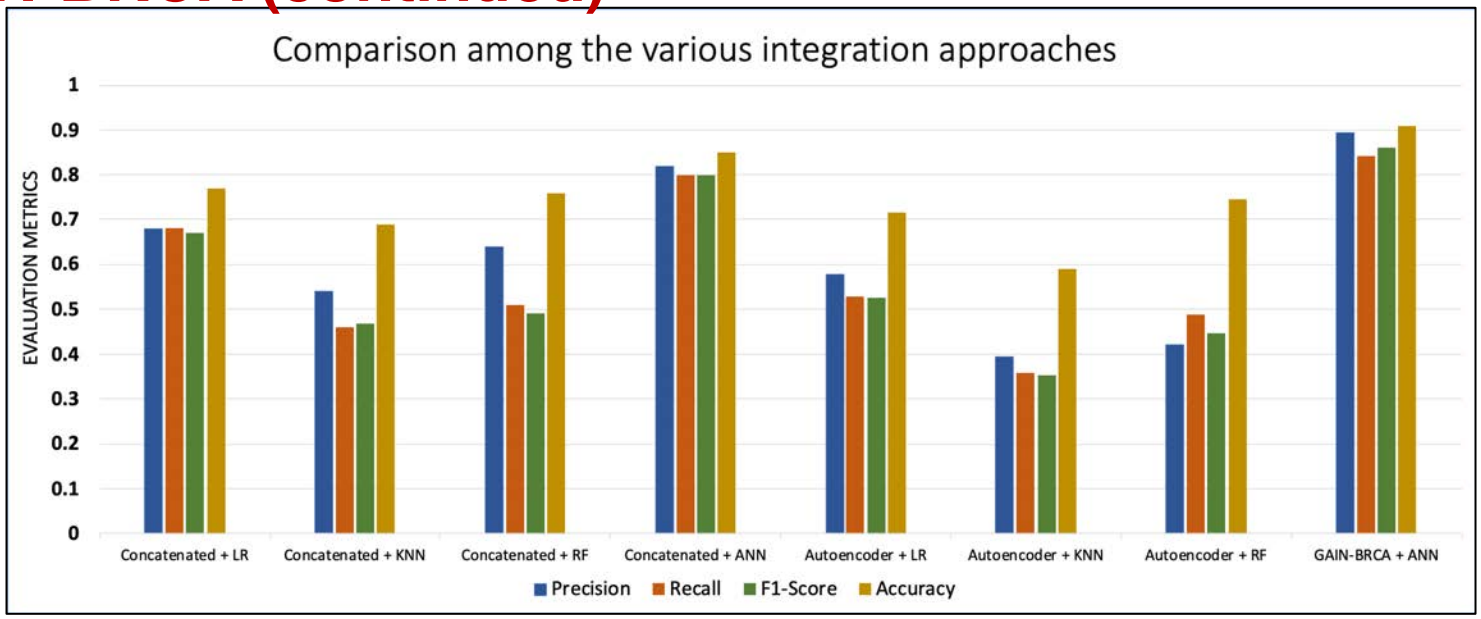


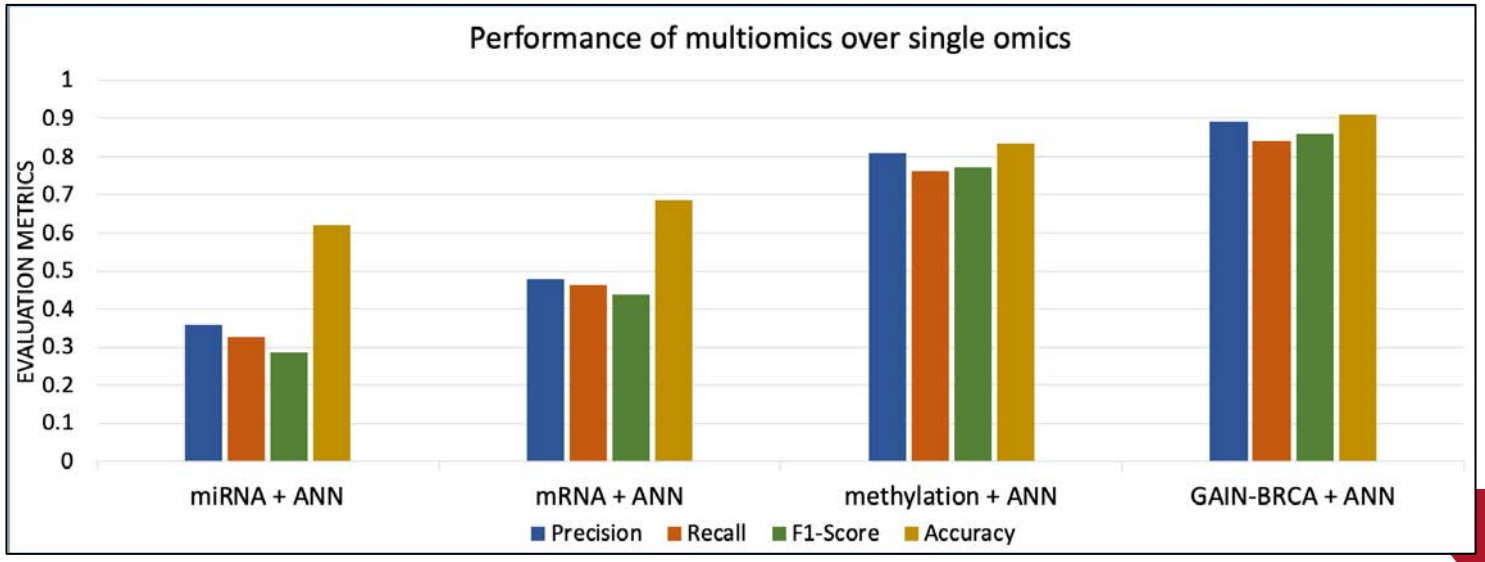
GAIN-BRCA (continued)

Deep-Learning based algorithmic strategy

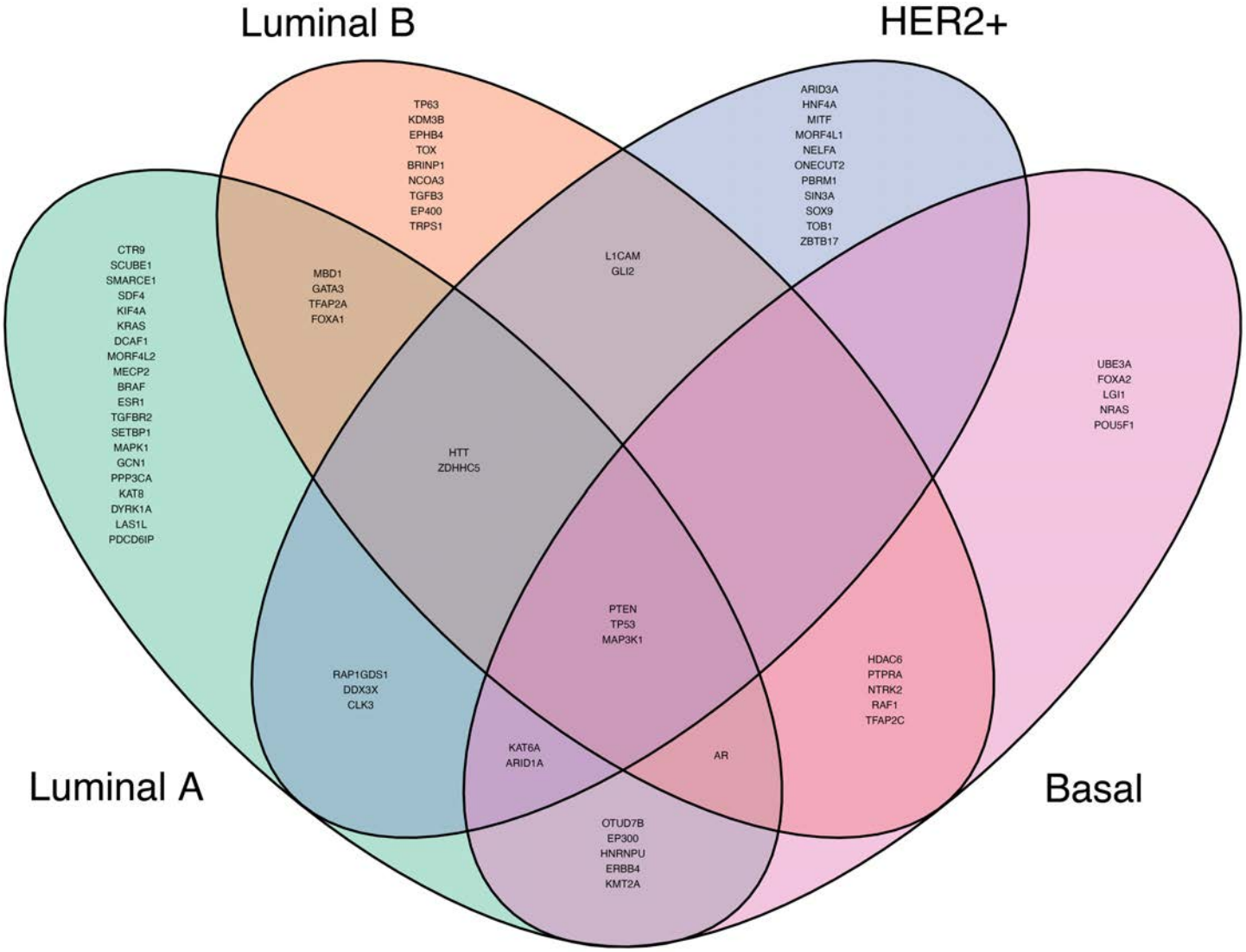


GAIN-BRCA (continued)



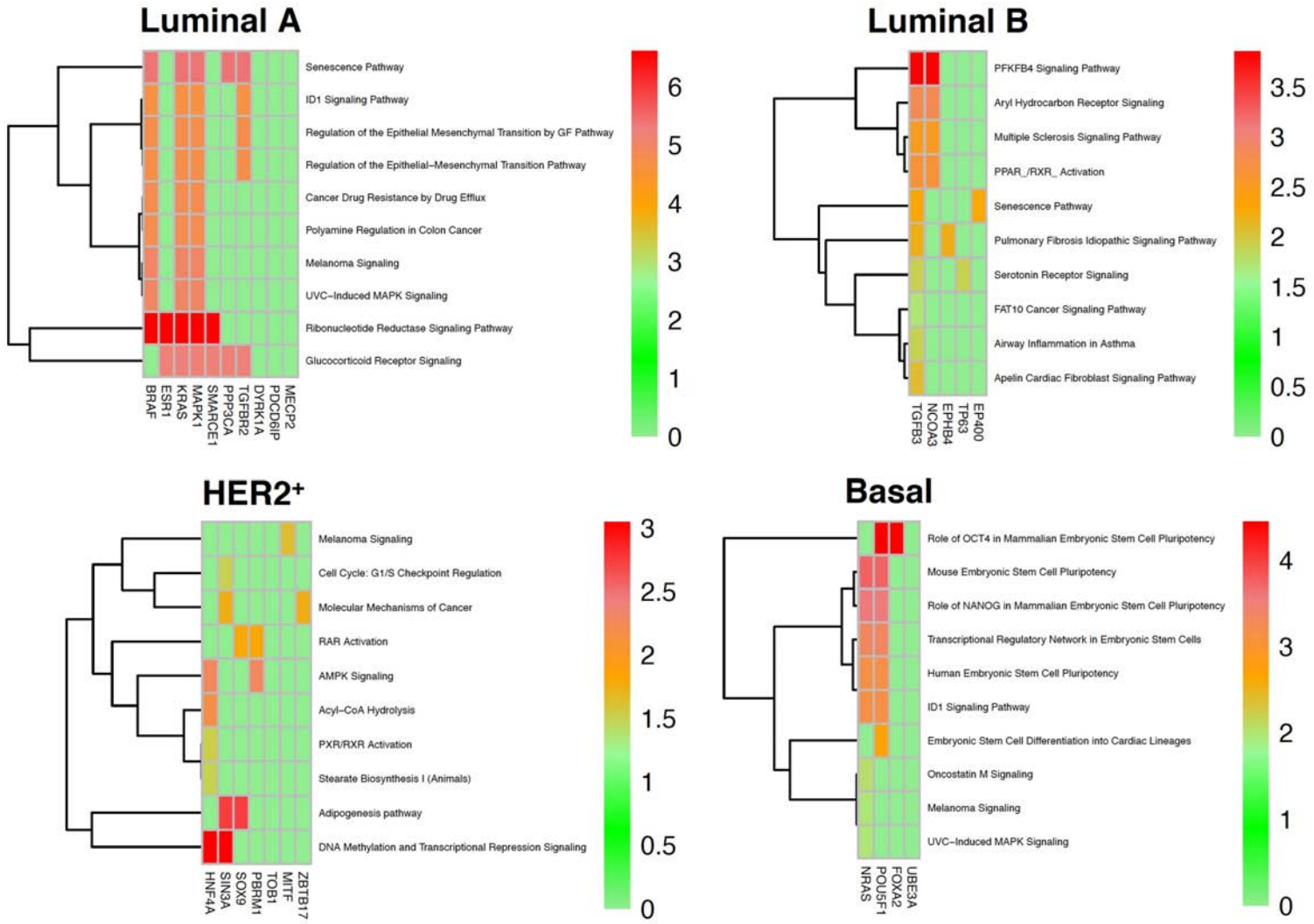


GAIN-BRCA (continued)



GAIN-BRCA (continued)

Top 10 enriched pathways in each subtype from IPA analysis



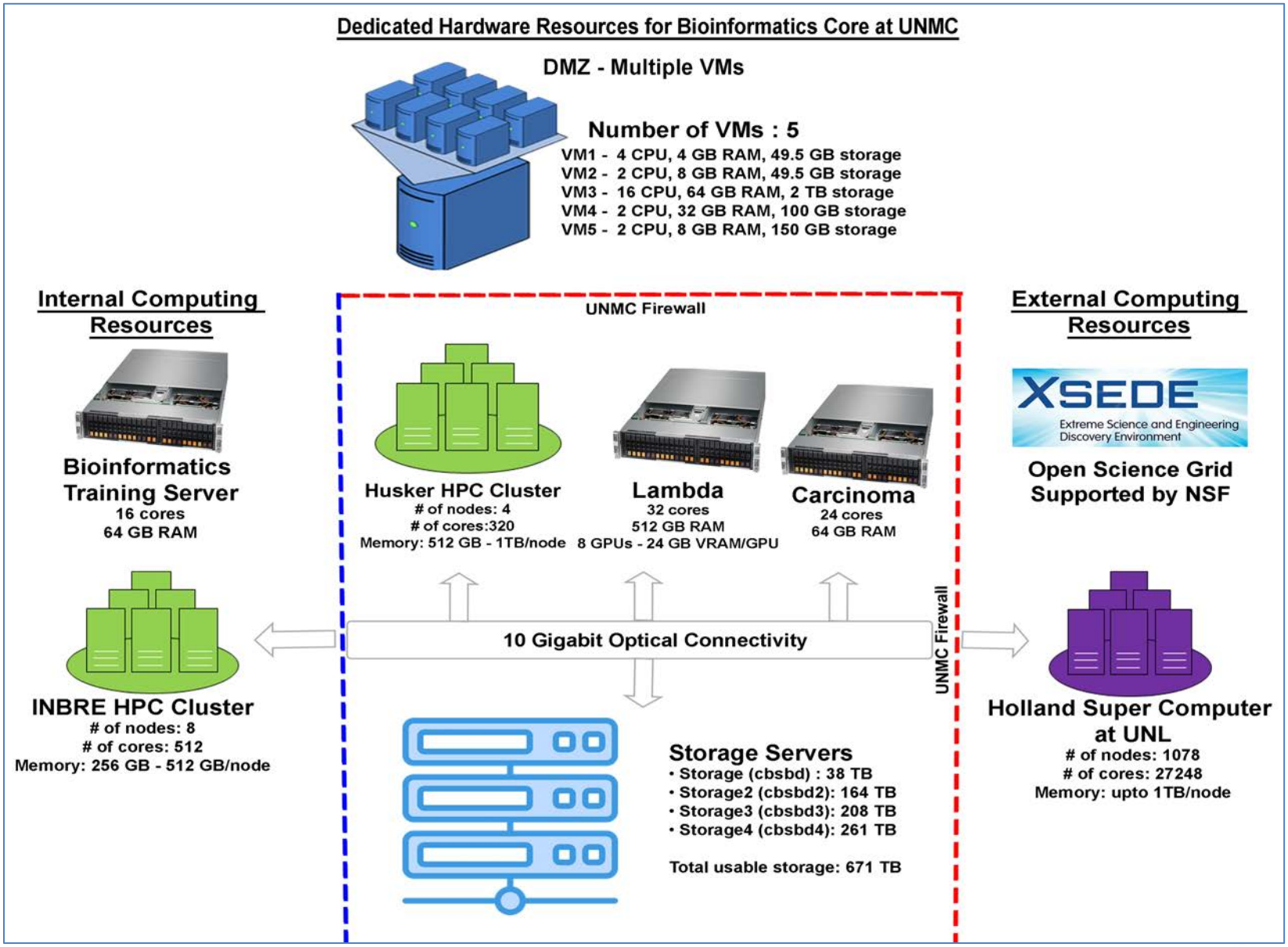
Building a Computational Ecosystem at UNMC

Components of the Ecosystem

- Computational infrastructure
  - Development servers
  - High-performance clusters
  - Virtual machines
  - Storage servers
- Tools and pipelines
  - Open-sources tools
  - Pipelines
- Genomic Data
  - TCGA
  - All-of-Us
- Personnel expertise
  - Programmers
  - Biologists
  - Bioinformaticians

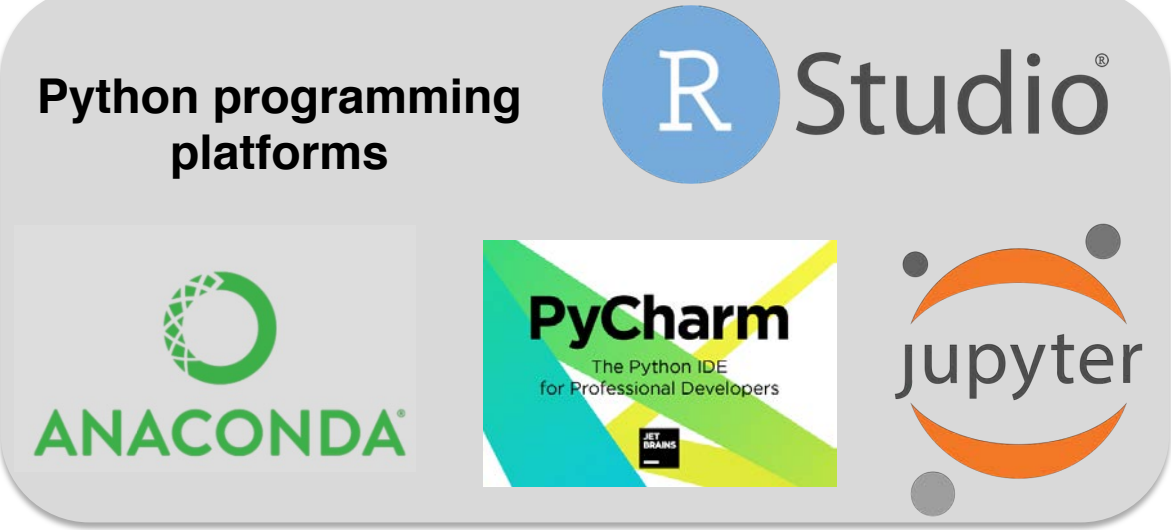
Locally installed open-source tools

Open-Source Tools	Description
ALLPATHS-LG	Whole-genome shotgun assembler
ANNOVAR	Tool for functional annotation of genetic variants
Bcbio-nextgen	Validated and scalable resource with variant calling and multiple NGS data analysis tools
BWA	Maps sequence reads to reference genomes
Cell Ranger	Software for the analysis for scRNA-seq data
ChimeRScope	Fusion transcript detection tool using RNA-seq data
FASTQC	A quality control tool for NGS data
FASTX-Toolkit	Command line tools to process FASTA/FASTQ files
GATK	The Genome Analysis Toolkit for NGS data analysis
I-TASSER	A protein 3-D structure prediction & modeling tool
MOFA	Unsupervised integration of multi-omics data sets
Monocle	A toolkit for pseudotime, clustering and differential expression analysis of single cell data
MuTect	Identifies somatic point mutations in NGS data
NetMHCpan & NetMHCIIpan	Linux-based platform to predict epitope binding to MHC I and MHC II molecules
NeoPredPipe	High throughput neoantigen prediction and recognition potential pipeline
OptiType	Precision HLA typing tools from NGS data
QIIME2	A powerful, extensible, and decentralized microbiome analysis package
SAMtools	Utilities for manipulating alignments including sorting, merging, indexing and formatting
Seurat 3.0	Tool for analysis/exploration of scRNA-seq data
STAR	Aligns RNA-seq reads to a reference genome
Trinity	Trinity is a de novo transcriptome assembler
Tuxedo Suite	Consists of Bowtie, Tophat, and Cufflinks, used in the RNASeq analysis pipeline
VirtualFlow	Ligand preparation and virtual screening workflow


Dedicated Hardware Resources for Bioinformatics Core at UNMC
DMZ - Multiple VMs
Number of VMs : 5
VM1 - 4 CPU, 4 GB RAM, 49.5 GB storage
VM2 - 2 CPU, 8 GB RAM, 49.5 GB storage
VM3 - 16 CPU, 64 GB RAM, 2 TB storage
VM4 - 2 CPU, 32 GB RAM, 100 GB storage
VM5 - 2 CPU, 8 GB RAM, 150 GB storage
Internal Computing Resources
UNMC Firewall
External Computing Resources
Bioinformatics Training Server
16 cores
64 GB RAM
Husker HPC Cluster
# of nodes: 4
# of cores:320
Memory: 512 GB - 1TB/node
Lambda
32 cores
512 GB RAM
8 GPUs - 24 GB VRAM/GPU
Carcinoma
24 cores
64 GB RAM
XSEDE
Extreme Science and Engineering Discovery Environment
Open Science Grid Supported by NSF
10 Gigabit Optical Connectivity
UNMC Firewall
INBRE HPC Cluster
# of nodes: 8
# of cores: 512
Memory: 256 GB - 512 GB/node
Storage Servers
• Storage (cbsbd) : 38 TB
• Storage2 (cbsbd2): 164 TB
• Storage3 (cbsbd3): 208 TB
• Storage4 (cbsbd4): 261 TB
Total usable storage: 671 TB
Holland Super Computer at UNL
# of nodes: 1078
# of cores: 27248
Memory: upto 1TB/node

Machine Learning: Tools & Frameworks

Python programming platforms



Machine Learning Library

Feature engineering tools
(selection/transformation)

Autoencoders

Google cloud

Google Cloud

Data Sizes and Processing times for various OMICS datasets

Data Type	File size/sample	Processing time
Whole Genome Sequencing	~50 - 80 Gigabytes	4 – 5 days
Whole Exome Sequencing	~10 - 20 Gigabytes	3 days
Bulk RNA Sequencing	~8 - 10 Gigabytes	2 days
Single cell RNA Sequencing	~3 - 5 Gigabytes	4 days
Bulk ATAC Sequencing	~8 - 10 Gigabytes	3 days
Single cell ATAC Sequencing	~3 - 5 Gigabytes	4 days

- Computing Infrastructure: High-performance clusters, batch submission, running slurm, etc.
- Storage: Requires Terabytes to Petabytes of storage capacity
- Network Bandwidth: Downloading and moving high volume data between servers
- IT Personnel: System administrators, data security, programmers
- Bioinformatics expertise: Working with appropriate tools, debugging, optimization, etc.

Challenges in the Genomic Big Data Analysis

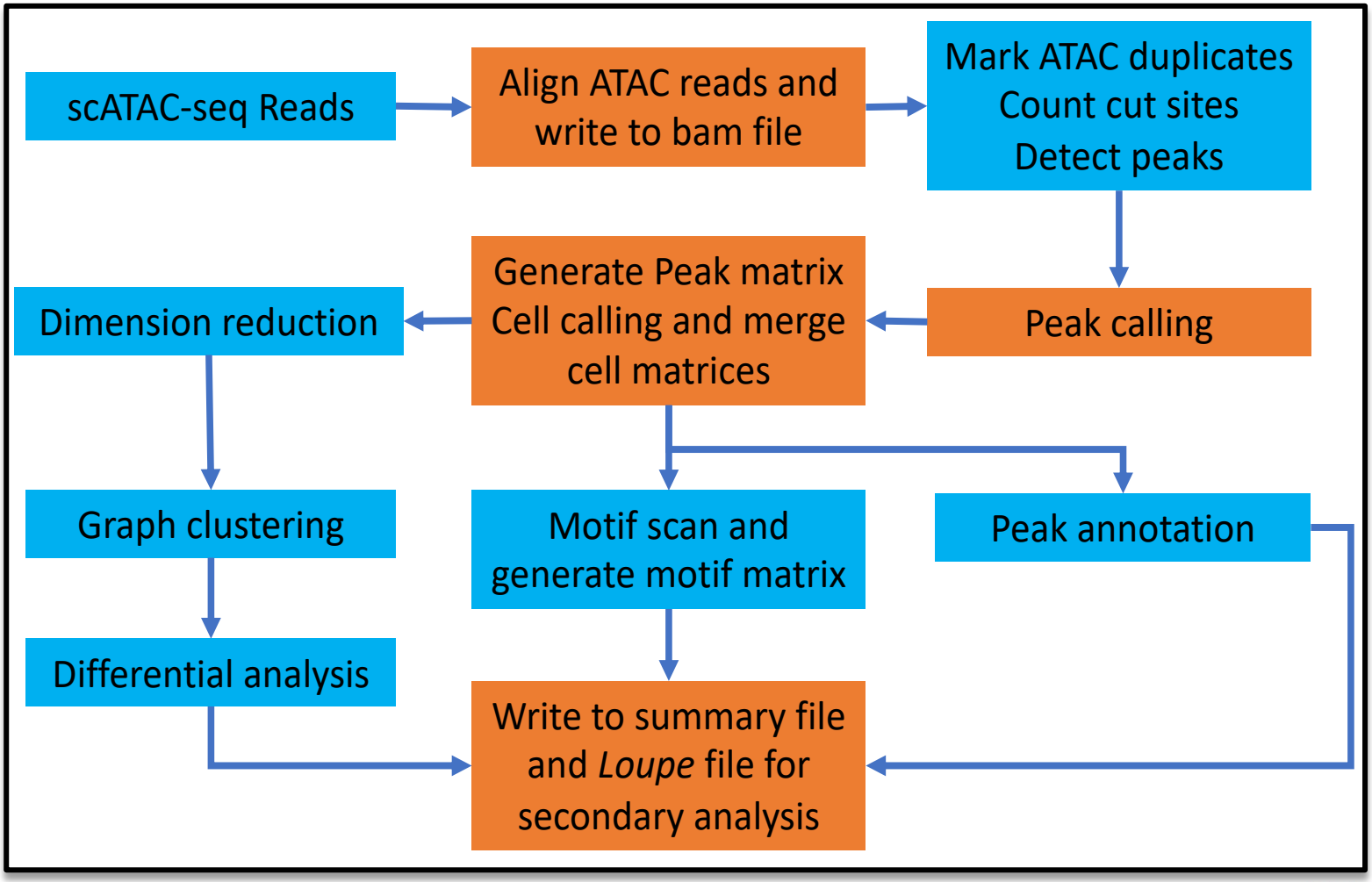
- Next-Generation Sequencing (NGS) datasets are very big
- Individual components are not effectively utilized unless the entire ecosystem is present.
- Small to medium research institutions can't afford to build robust infrastructure

- Cloud-based learning modules offer a solution to this problem
  - The computational ecosystem with data and tools is maintained on the cloud.
  - Cloud-based modules can be accessed from anywhere with internet access, which democratizes access to all institutions.

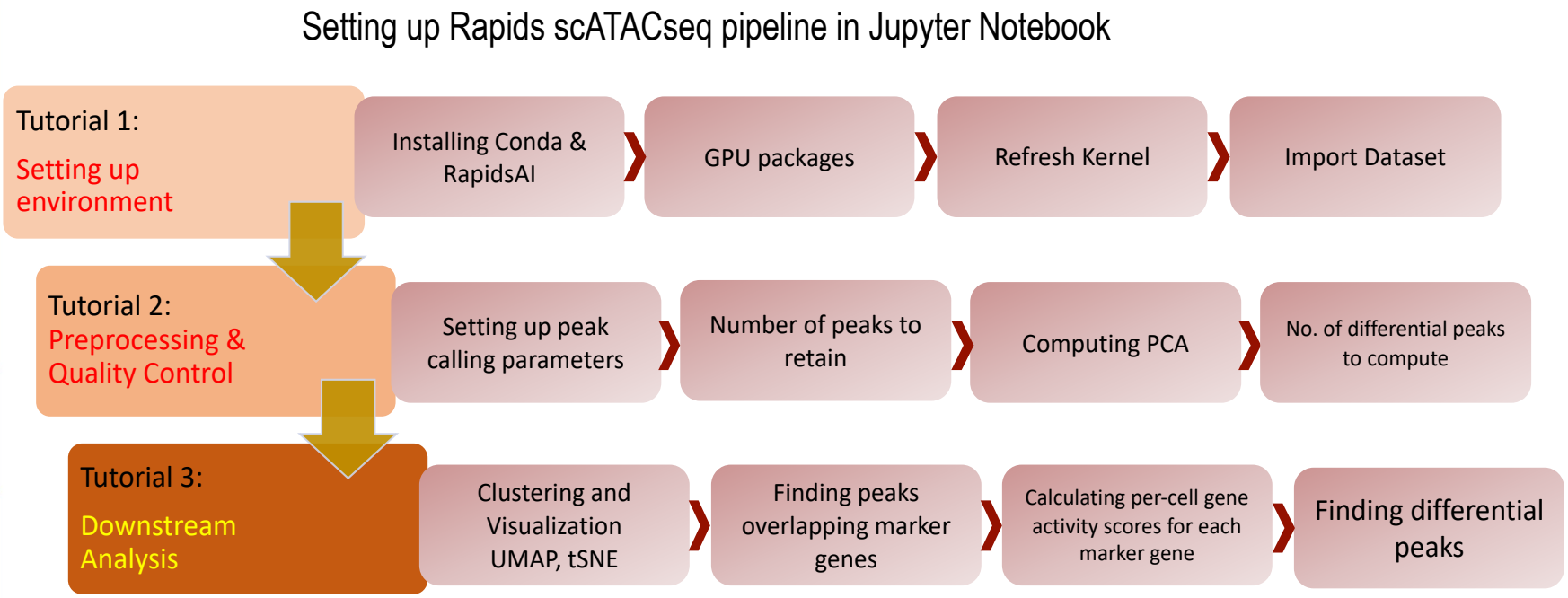
Cloud-ATAC: a self-learning module for single cell ATAC-seq Data Analysis on the Google Cloud Platform

(NIH/NIGMS - 5P20GM103427 (NE-INBRE)

Single Cell ATAC-seq Data Analysis - Workflow



Single Cell ATAC-seq Data Analysis – A Cloud-based Learning Module



Programming Environment and Tools Used

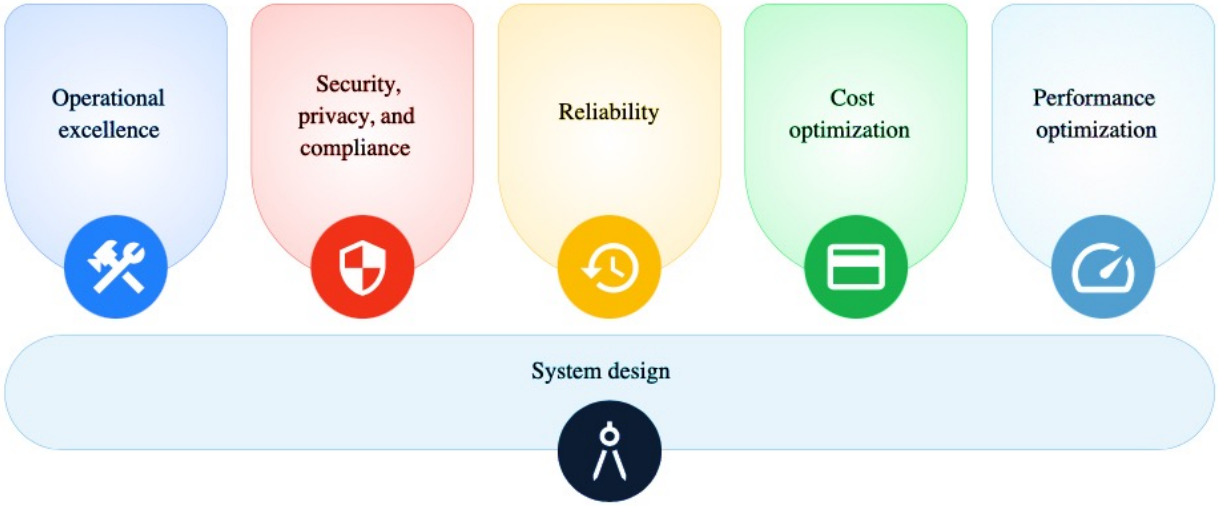
- Environment
  - Google Cloud Platform
  - Rapids AI package
- Programming Languages
  - Python
  - Implemented in Jupyter Notebooks
  - Quizzes and Flash cards stored in json files
- Data Analysis Tools
  - QC & Pre-Processing:
    - Such as Trimmomatic, MultiQC, Picard, Samtools
  - Mapping, Clustering, and Downstream Analysis:
    - Such as BWA, Deeptools, UMAP, tSNE, TOBIAS, Homer

Google Cloud Platform (GCP) Environment

Google Cloud
Vertex Clusters

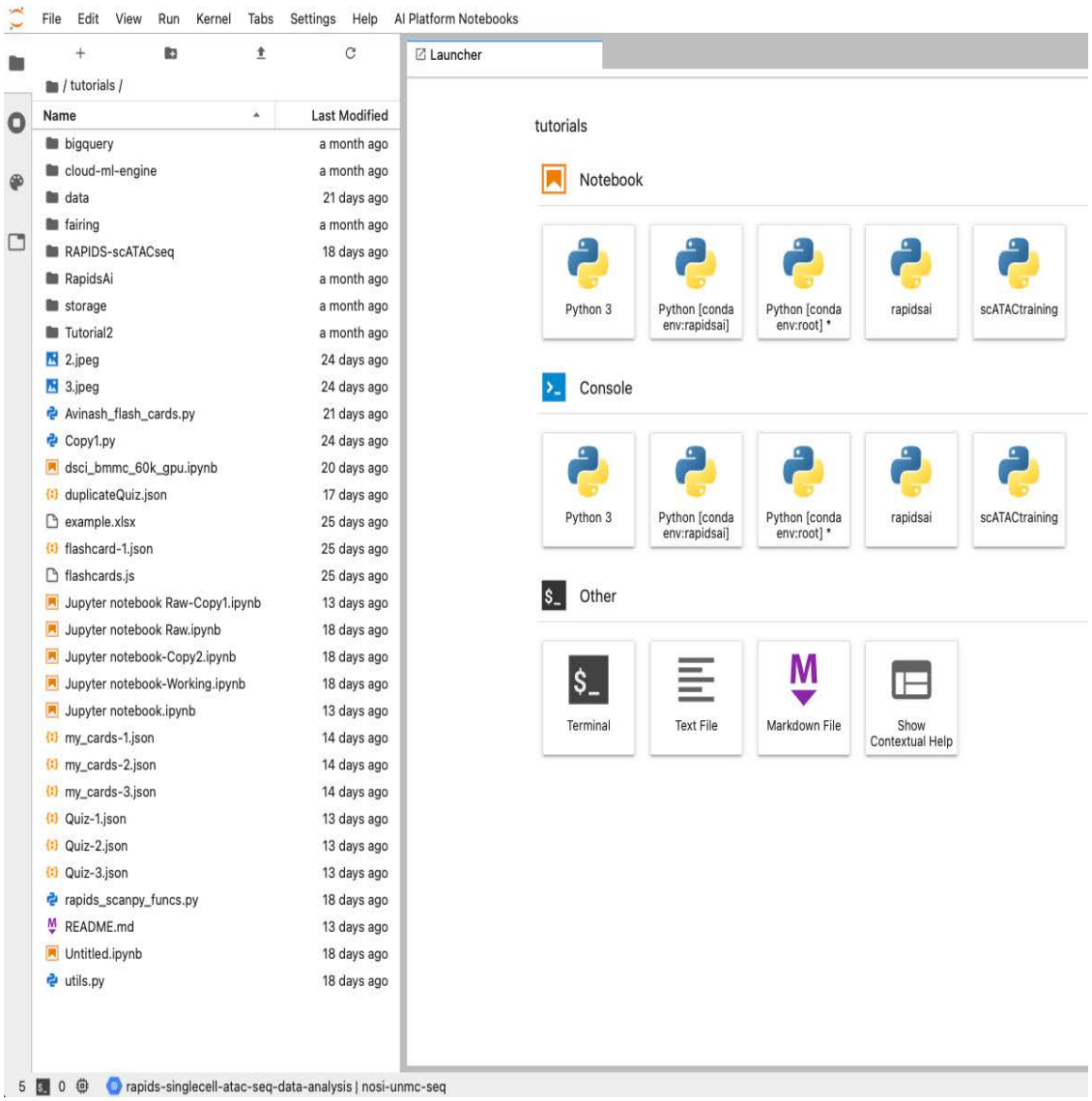
Open-sourced OMICS Pipelines

The Google Cloud Architecture Framework is organized into six categories, as shown in the following diagram:

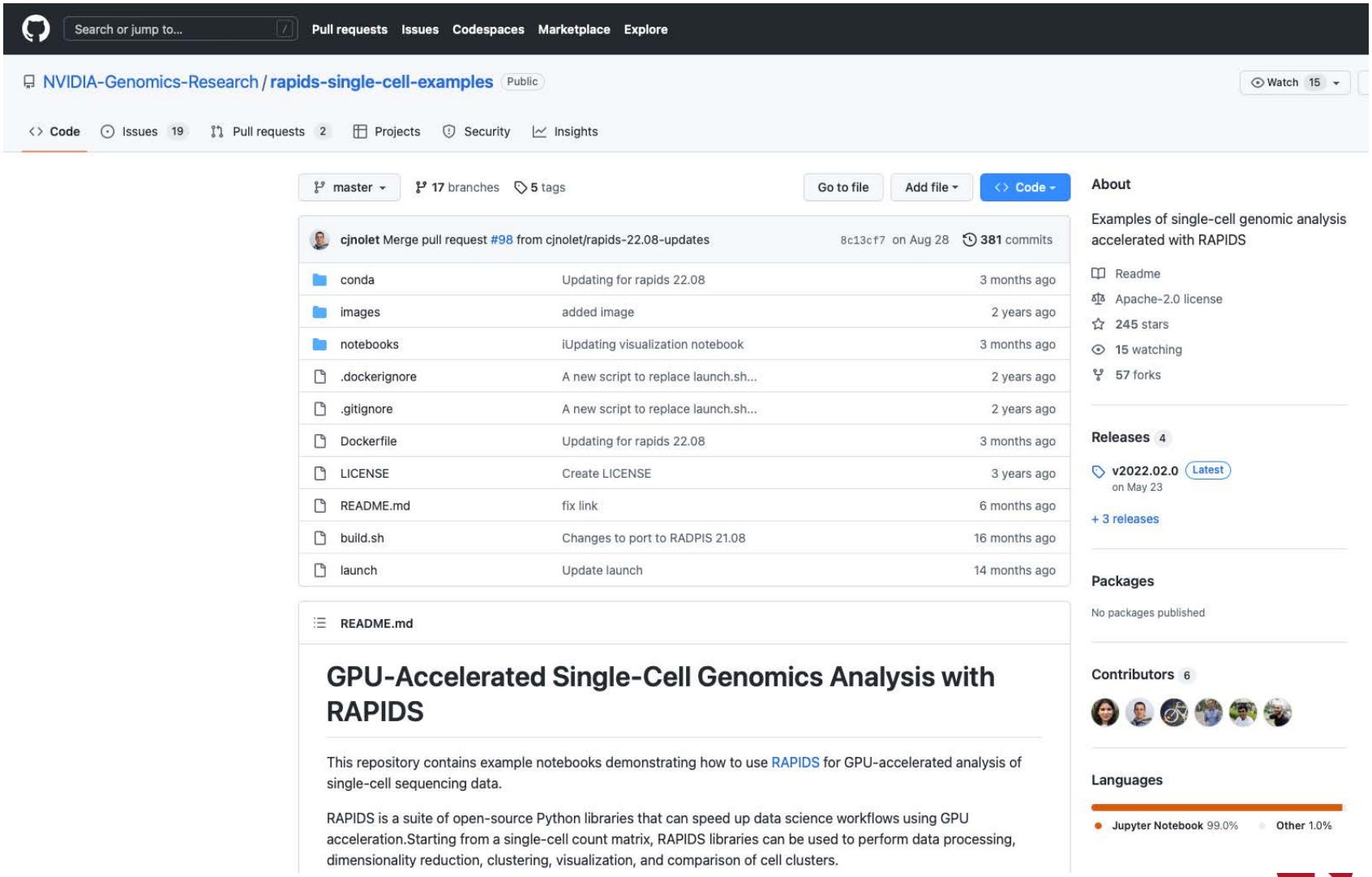


Jupyterhub

- Multilingual command usage.
  - Python
  - R
  - HTML
- Supplements OMICS pipeline with learning components such as, flash cards, quizzes and videos.
- Easy integration with other AI models
  - Ex. RapidsAI



Running scATAC-seq Analysis Pipeline Using RAPIDS-AI



Using Example Dataset: Droplet Single-cell ATAC-seq of 60K Bone Marrow Cells taken from Lareau et al., Nat Biotech 2019

Setting up Conda and RAPIDS-AI Environment

STEP 1a: Run the following commands in TERMINAL

```
#To access terminal
#follow this path File > New > Terminal
#Then do Step 1b to 1f in Terminal
#Do step 1g in Jupyter notebook
```

```
# 1b Create conda environment for rapids, then activate it
conda create -n rapidsai
conda activate rapidsai
```

```
# 1c install rapids
conda install -c rapidsai -c nvidia -c conda-forge python=3.7 cudatoolkit=11.2 rapids=21.12 llvmlite gcsfs openssl dask-sql
```

```
# 1d install scanpy and wget
pip install scanpy wget
```

```
# 1e add rapidsai environ
python -m ipykernel install --user --name=rapidsai
jupyter kernelspec list
```

```
# 1f download required files
wget https://raw.githubusercontent.com/NVIDIA-Genomics-Research/rapids-single-cell-examples/master/notebooks/utils.py
wget https://raw.githubusercontent.com/NVIDIA-Genomics-Research/rapids-single-cell-examples/master/notebooks/rapids_scanpy_funcs.py
wget https://raw.githubusercontent.com/NVIDIA-Genomics-Research/rapids-single-cell-examples/master/notebooks/dsci_bmmc_60k_gpu.ipynb
```

```
# 1g open notebook and change kernel in jupyter notebook to use rapidsai envs
# jupyter menu: goto > kernel > change kernel > rapidsai
```

```

```

```
# create conda envs
#run these two lines on terminal
conda create -n rapidsai
conda activate rapidsai
```

```
#run this line on terminal
conda install -y --prefix /usr/local -c rapidsai -c nvidia -c conda-forge python=3.7 cudatoolkit=11.2 rapids=21.12 llvmlite gcsfs openssl dask-sql
```

```
#run this line on terminal
python -m ipykernel install --user --name=rapidsai
```

```
#run this line on terminal
jupyter kernelspec list
```

```
#in jupyter notebook, do the following.
Click on jupyter menu: goto > kernel > change kernel > rapidsai
```

QC: Plots to Visualize per-base Sequencing Coverage

Plot per-base sequencing coverage

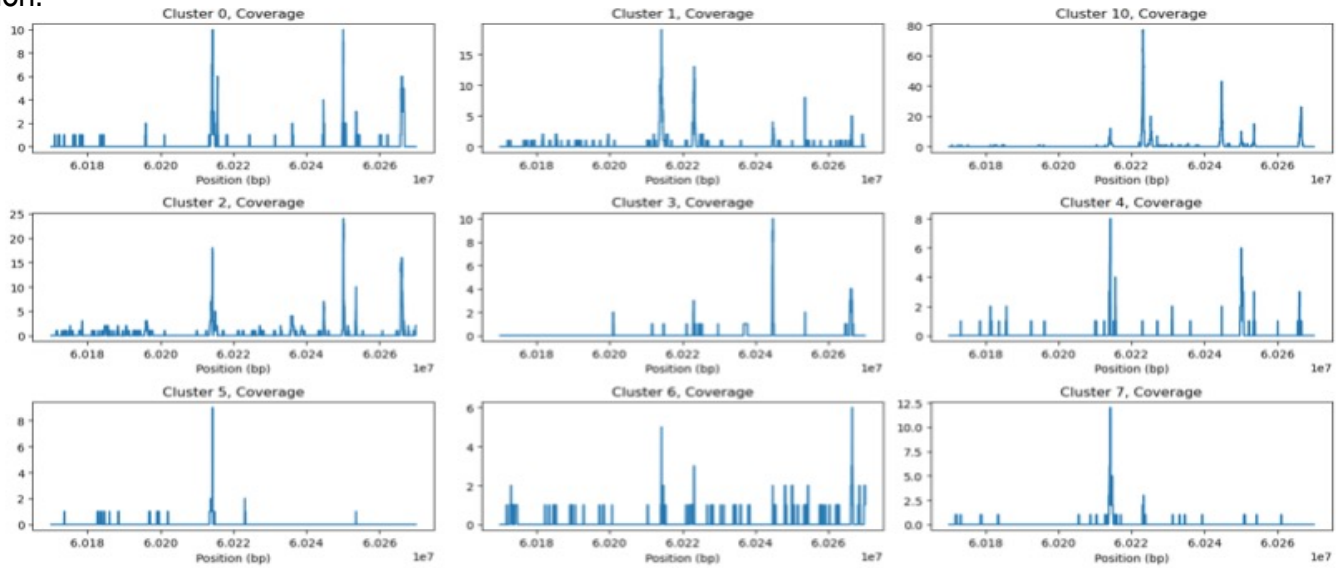
Below, we visualize this unsmoothed measure of chromatin accessibility peaks across selected region.

```
In [25…

fig, axs = plt.subplots(ncols = 3, nrows = 4, figsize = (16, 10), dpi=100)
axs = axs.flatten()
coords = np.arange(start, end)

for (i, cluster) in enumerate(cluster_ids):
    axs[i].plot(coords, region_coverage[i,:])
    axs[i].set_title(f'Cluster {cluster}, Coverage')
    axs[i].set_xlabel('Position (bp)')

plt.tight_layout()
plt.show()
```



Graph-based Clustering Using Louvain and Leiden Methods

[15]:
```
%%time
adata.obs['leiden'] = rapids_scanpy_funcs.leiden(adata)
```

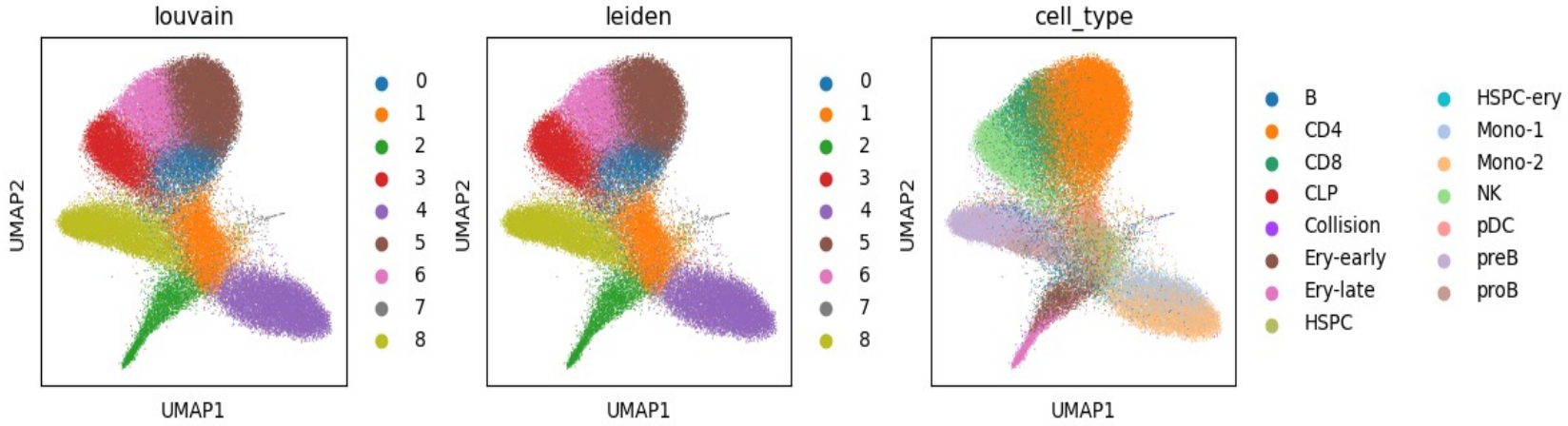
```
CPU times: user 92.3 ms, sys: 42.6 ms, total: 135 ms
Wall time: 156 ms
```

[16]:
```
%%time

fig, axs = plt.subplots(nrows = 1, ncols = 3, figsize = (12, 3), dpi=100)

sc.pl.umap(adata, color=['louvain'], ax=axs[0], show=False)
sc.pl.umap(adata, color=['leiden'], ax=axs[1], show=False)
sc.pl.umap(adata, color=['cell_type'], ax=axs[2], show=False)

plt.tight_layout()
plt.show()
```




```
CPU times: user 1.82 s, sys: 361 ms, total: 2.18 s
Wall time: 1.88 s
```

Visualizing the Transcription Factor Bound Motifs Using IGV

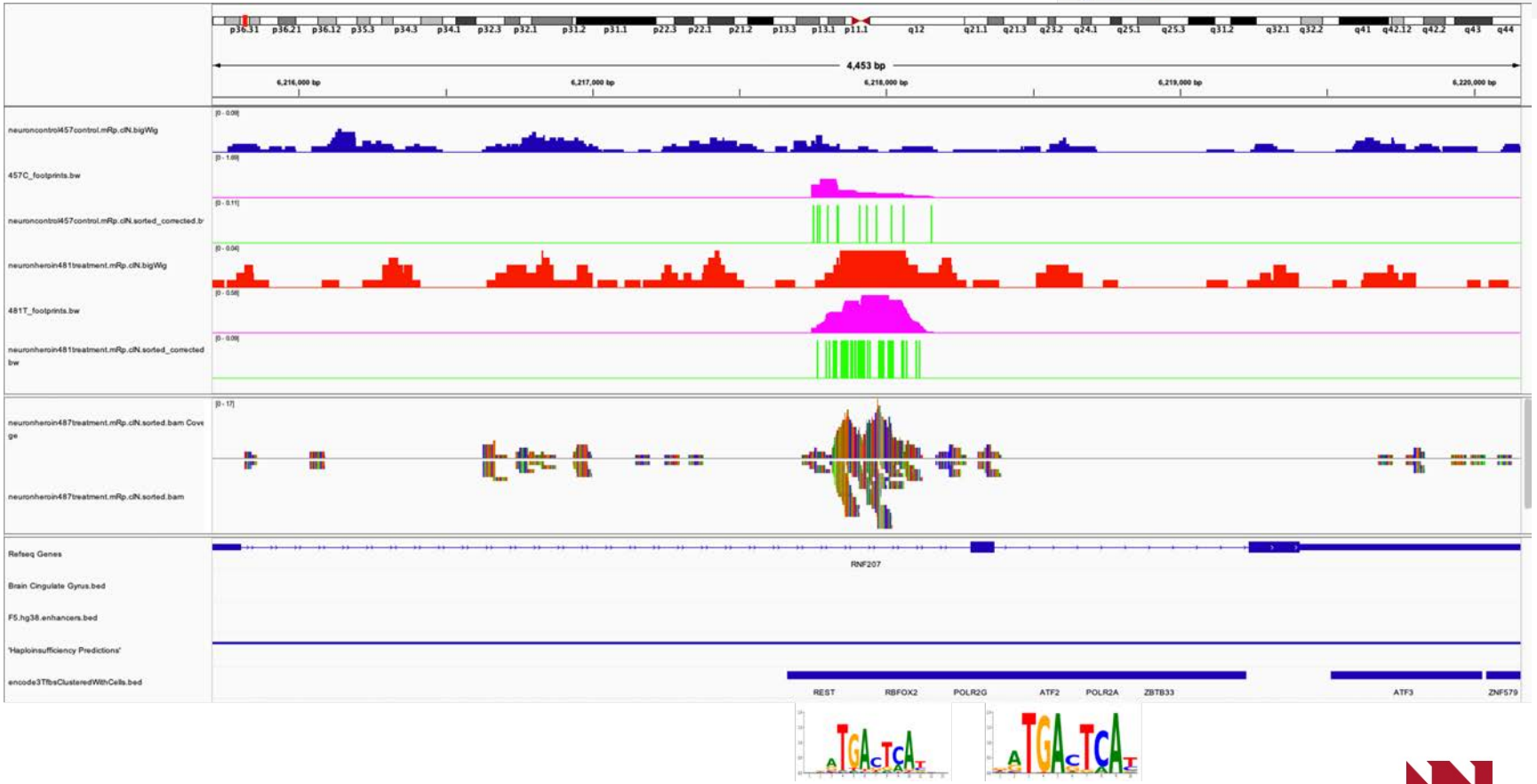
Genome Browser

Now that we have our bigwig files, we can visualize the signal in a genome browser. We'll use igv in this example.

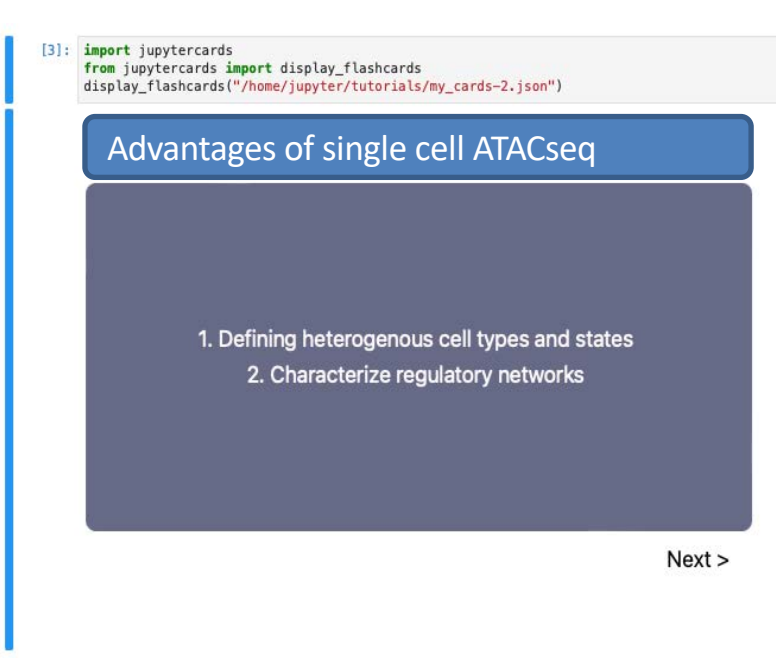
In [43]:

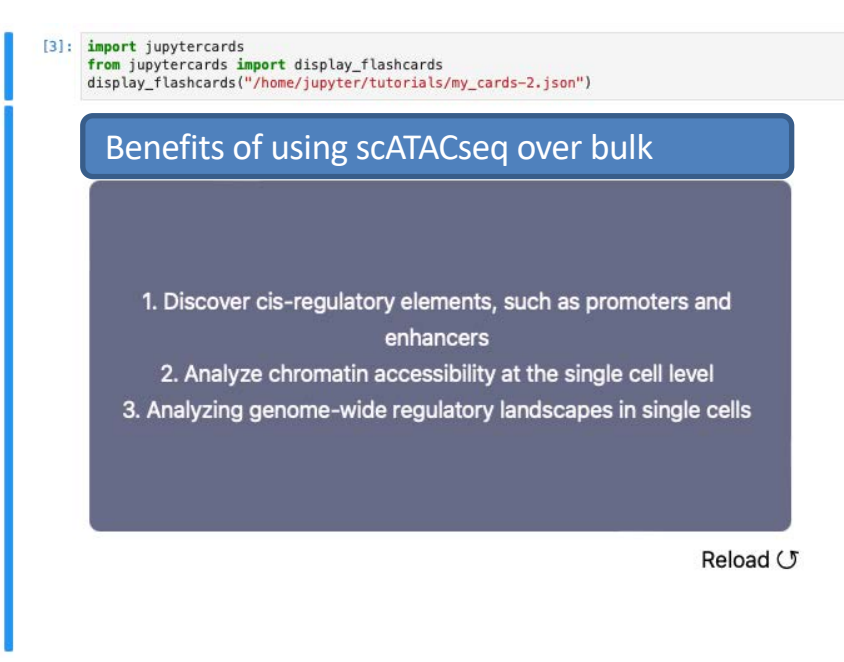
```
igv_notebook.init()
myigv = igv_notebook.Browser(
    {
        "genome": "hg38",
        "locus": "chr4:55,400,000-56,400,000"
    }
)
myigv.load_track(
{
        "name": "CTL",
        "url": "Tutorial2/BigWigFiles/Control.bw",
        "format": "bigwig",
        "type": "wig"
    }

)
myigv.load_track(
{
        "name": "Mutant",
        "url": "Tutorial2/BigWigFiles/Mutant.bw",
        "format": "bigwig",
        "type": "wig"
    }
```



Using Flashcards for Enhanced Learning





Using Quizzes to Evaluate Learning Outcomes

```
[1]: import jupyterquiz
     from jupyterquiz import display_quiz
     display_quiz("/home/jupyter/tutorials/Quiz-1.json")
```

ATAC-seq protocols have struggled with an average of 50% - 80% contaminating mitochondrial DNA reads and are removed during processing. The mitochondrial genome has higher accessiblity due to?

no ATAC-seq peaks of interest	lack of chromatin packaging
lesser number of genes	lack of protein-coding genes

Not quite. The mitochondrial genome contains 37 genes that encode 13 proteins, 22 tRNAs, and 2 rRNAs

```
[1]: import jupyterquiz
     from jupyterquiz import display_quiz
     display_quiz("/home/jupyter/tutorials/Quiz-1.json")
```

ATAC-seq protocols have struggled with an average of 50% - 80% contaminating mitochondrial DNA reads and are removed during processing. The mitochondrial genome has higher accessiblity due to?

lack of chromatin packaging	no ATAC-seq peaks of interest
lesser number of genes	lack of protein-coding genes

Yes. The mitochondrial genome is not enveloped, and is not packaged into chromatin

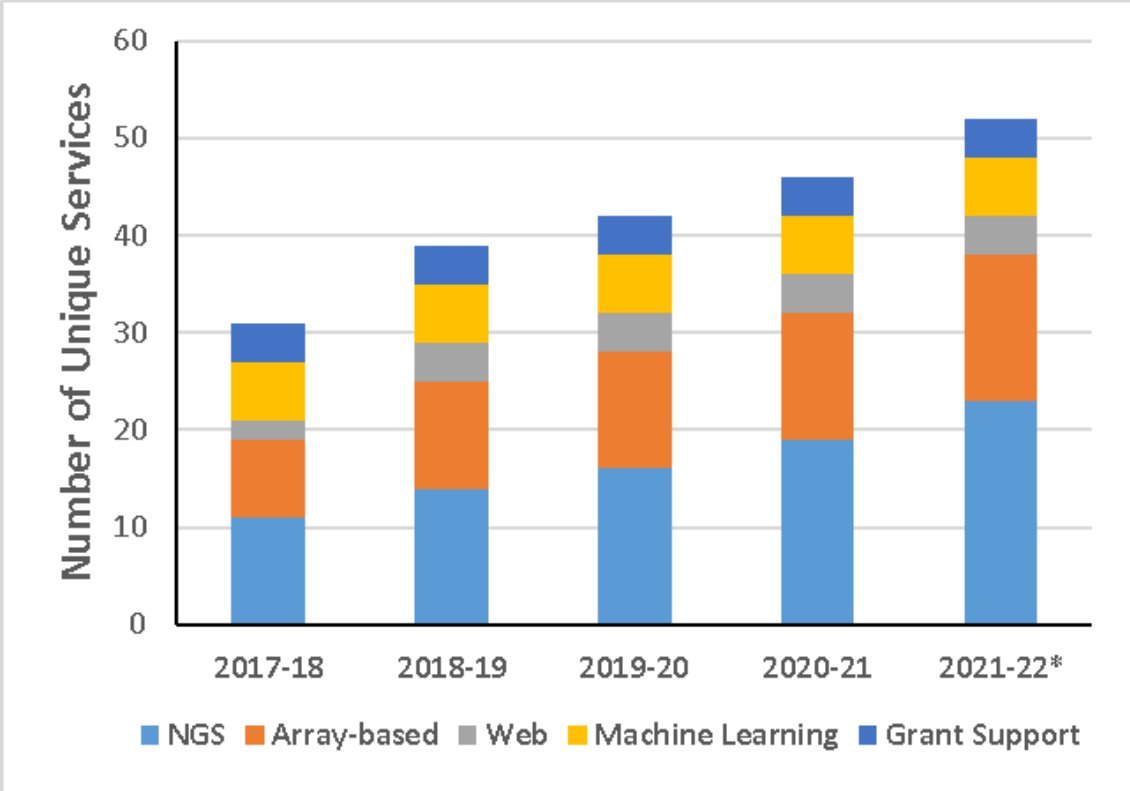
Bioinformatics and Systems Biology Core

Bioinformatics and Systems Biology Core

Services Offered

- NGS (Next-Generation Sequencing) Data Analyses
  - RNAseq and miRNAseq
  - Exome and whole genome seq; chip-seq; proteomics
  - Single-cell seq and single-cell ATAC-seq
  - DNA methylation seq
  - Metagenomics seq
  - De novo assembly and annotation
  - GWAS (Genome Wide Association Studies) and PLINK analysis
  - Alternative splicing analysis for RNAseq
  - Cell free DNA analysis to identify tumor purity
  - Nucleotide percentages in ORFs; genomic ribosome binding sites identification
- System Biology Data Services
  - Ingenuity Pathway Analysis (IPA) and gene pathway and network analyses
  - Functional analysis of gene sets using GSEA, DAVID, ClueGO, KEGG, etc.
  - Copy-number variation analysis for CGH arrays; DNA methylation arrays
  - Survival analysis and plotting,volcano, heatmap, and Venn diagram plotting
  - TCGA 'Omics' data analyses and NCBI/GEO array data analysis
  - Oncomine and cBioPortal data analyses
  - SNP association analysis for SNP array
  - Metabolomic pathway and integrated pathway analysis
  - miRNAseq target gene pathway; lncRNAs targeting gene promoters and mRNA
- Web Applications and Databases
  - Development of web sites and web applications to publish research data
  - Development of searchable relational databases for complex research data
- Machine Learning and Research Computing
  - Clustering and modeling experimental data using machine learning
  - Development of custom programs to analyze complex data
  - Motifs/pattern discovery from large datasets
- Grant and Publication Support and Consultation
  - Pre-grant consultation, support letters, collaborations
  - Consultation on the NGS/Bioinformatics experimental design and budget quotes
  - Generate figures to represent high-dimensional data
  - Consultation on Data Management and Sharing (DMS) Plans
- Other Bioinformatic Related Support
  - Sharing data via ftp, depositing data into SRA/GEO/NCBI resources, setting custom packages like local BLAST, Galaxy or other servers.
- Analysis Tools: Ingenuity Pathway Analysis (IPA), CLC Genomics Workbench, Schrödinger, GROMACS, BioCyc, CASSAVA, Partek, MATLAB, GraphPad Prism, SnapGene, R packages, NGS analysis tools: Tuxedo Suite, bcbio-nextgen, Nextflow, etc.

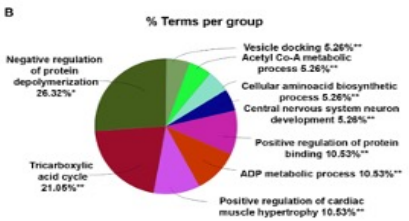
Bioinformatics Core Service Categories



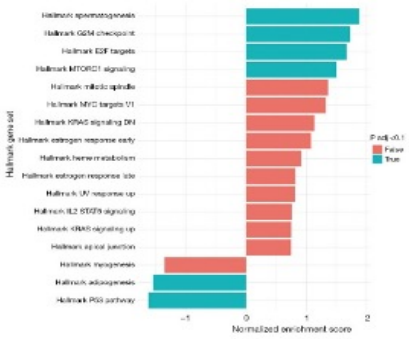
Licensed Software tools available

- Ingenuity Pathway Analysis (IPA)
- GraphPad Prism
- SnapGene
- CLC Genomics Workbench
- EndNote
- BioCyc Data Collection
- Machine learning software
    - Tenserflow
    - Keras
    - SciKit
    - Matplotlib

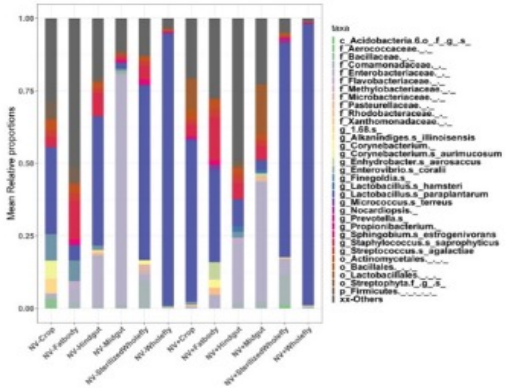
Hi-dimensional Data Analysis



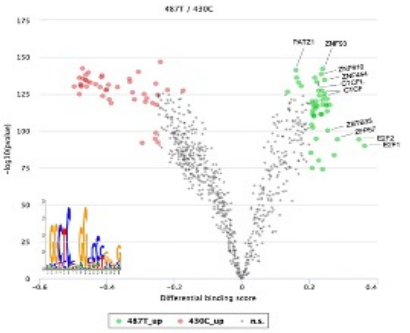
ClueGO analysis



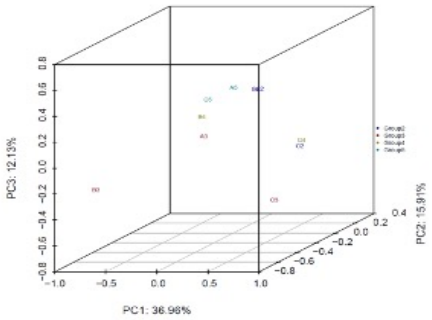
GSEA



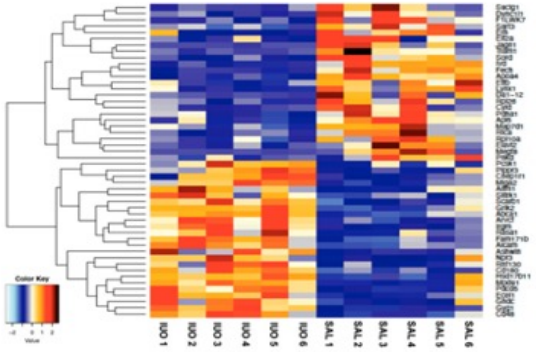
Metagenomics taxonomy



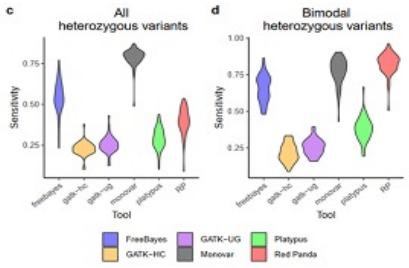
Volcano



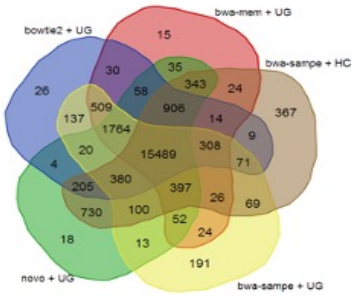
3D PCA



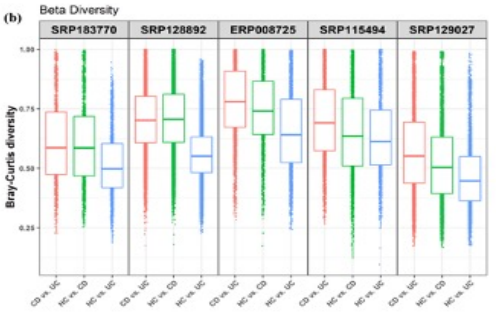
Heatmap



Violin plot

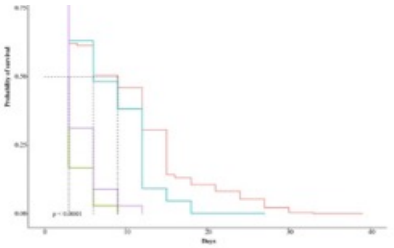


Venn diagram

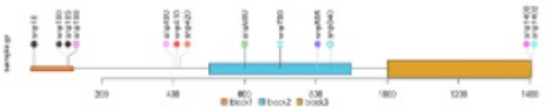


Box Plot

Hi-dimensional Data Analysis

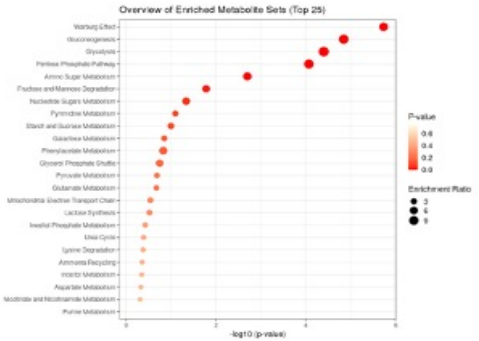


Survival plot

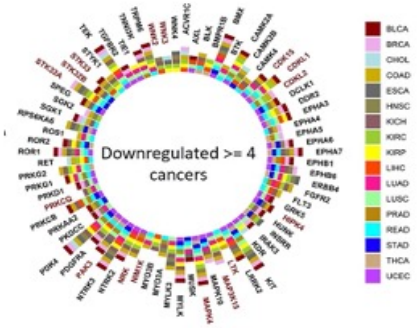


Lollypop plot

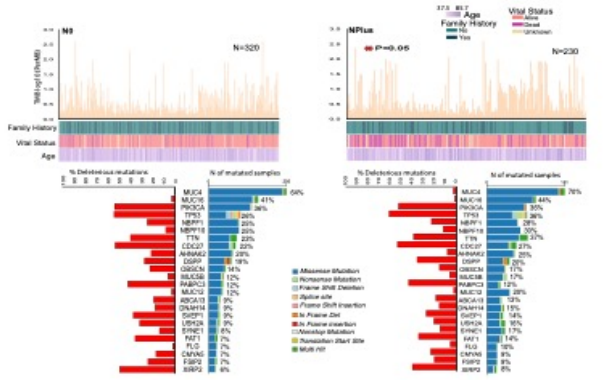
IPA match analysis



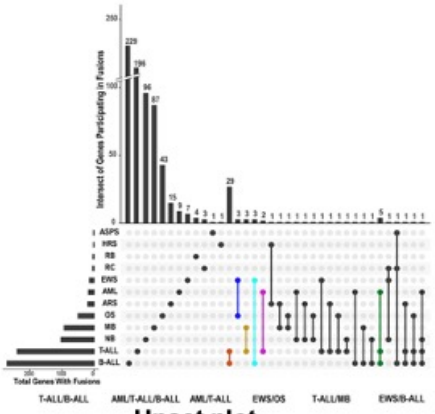
Metabolomic pathway bubble plot



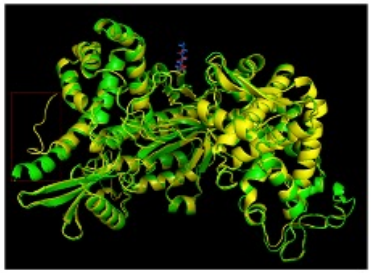
Circos plot



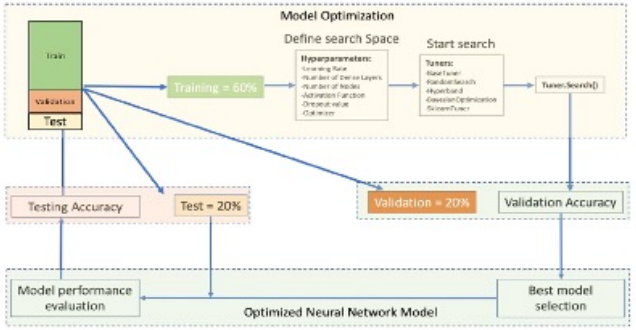
Genomic data mutational plot



Upset plot



Protein modeling



Machine Learning

Acknowledgements

Current Members
Peng Xiao, Assistant Professor
Sushil Shakyawar, Postdoc
Jai Patel, Postdoc
Meng Niu, Assistant Professor
Nagarajan Nagasundaram, Postdoc
Avinash Veerappa, Instructor
Subu Jagadesan, Bioinformatics Scientist
Rajashree Chakraborty, PhD candidate
Sahil Sethi, PhD candidate
Sujith Manavalan, PhD student

Former Members
Siddesh Southekal, PhD student
Nitish Kumar Mishra, Instructor
Suleyman Vural, PhD student
Brian King, PhD student

Funding: (NIH, DoD)
2P01AG029531
5P20GM103427 (NE-INBRE)
5U54GM115458 (IDeA-CTR)
5P30CA036727
NSRI Task Order FA4600-18-D-9001

- **Jordan Rowley – Assistant Professor, GCBA**
- **Google Cloud Platform**
- **Holland Computing Center, NU**
- **Bioinformatics and System Biology Core, UNMC**
- **Next-Gen Sequencing Core, UNMC**