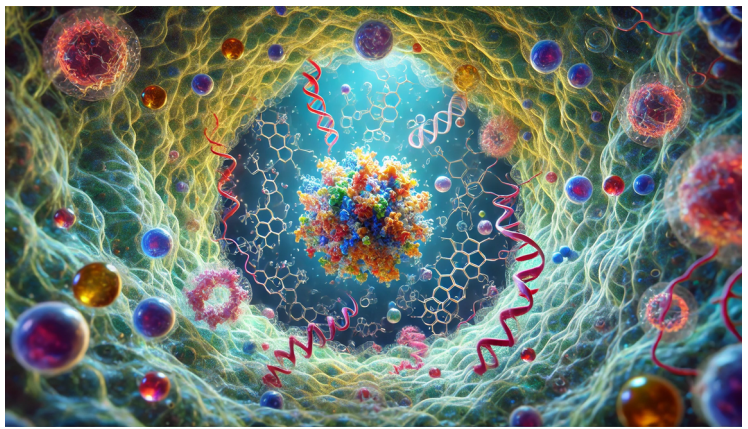
Statistical Approaches for ‘Omics Integration



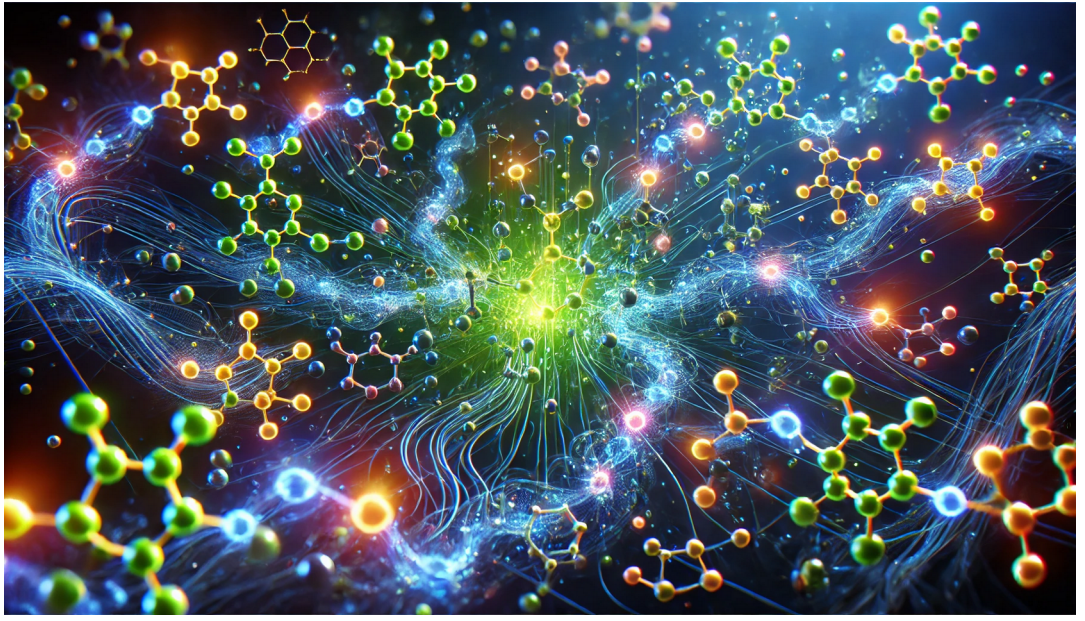
Joseph McElroy, Ph.D.

Lianbo Yu, Ph.D.

Slides available at:
https://github.com/omicsda/tutorial

Outline

- Introduction
- Preprocessing
- Methods
- Methods Focus
- Case Study
- Summary/Wrap-up/Questions



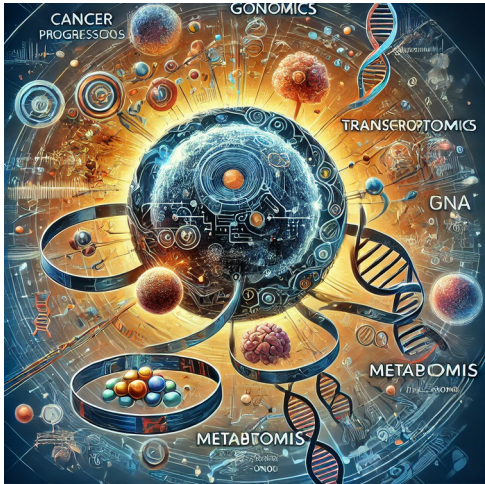
What is Multi-Omics Integration?

- The process of synthesizing and analyzing data from multiple 'omics' layers.
- The challenge of integrating heterogeneous, high-dimensional biological datasets (e.g., genomics, transcriptomics, proteomics) to derive meaningful insights.
- The application of computational, statistical, and machine learning methods to fuse and interpret multi-omics data while addressing batch effects, missing data, and high dimensionality.
- A paradigm that recognizes biology as an interconnected network rather than isolated molecular events, capturing regulatory and functional interdependencies between different omics layers.

Why Integrate Multi-Omics Data?

- Single-layer analyses miss critical interactions.
- Captures interdependencies among biological layers.

- Example: Cancer progression analysis linking:
  - Genetic mutations (genomics)
  - Gene expression shifts (transcriptomics)
  - Metabolic changes (metabolomics)



‘Omics Data

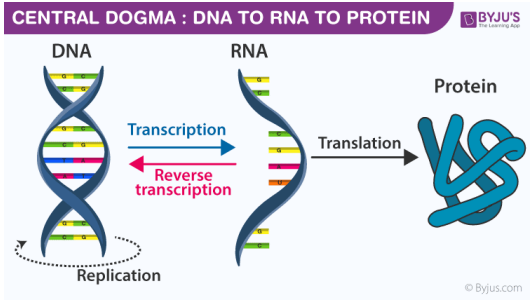
- Genomics: DNA-level information (SNPs, CNVs, mutations)

- Transcriptomics: RNA-level gene expression (mRNA, miRNA, lncRNA)

- Proteomics: Protein quantification & modifications

- Metabolomics: Profiling of metabolites

- Epigenomics: DNA methylation & histone modifications



https://byjus.com/biology/central-dogma-inheritance-mechanism/

Applications of Multi-Omics Integration

- Complex diseases involve multiple biological factors across omics layers.

- Applications:
  - **Precision Medicine:** Tailored treatments from multi-omics data.
  - **Disease Subtyping:** Identifying disease variants (e.g., breast cancer subtypes from TCGA).
  - **Biomarker Discovery:** Multi-layered approaches for predictive markers.
  - **Drug Repurposing:** Identifying new uses for existing drugs.

Key Challenges in Multi-Omics Integration

- **Heterogeneity Across Data Types**:
  - Different measurement platforms (e.g., RNA-seq vs. proteomics).
  - Inconsistent scales (e.g., counts vs. continuous data)
- **High Dimensionality**:
  - Large number of features (>20,000 genes, 500,000 CpG sites) vs. limited samples.
- **Batch Effects**:
  - Systematic differences from lab variations.
  - Solutions: Batch correction (ComBat, sva).
- **Missing Data**:
  - Imputation techniques?: kNN, Bayesian, matrix factorization.

Goals of This Tutorial

- **Conceptual Understanding**:
  - Fundamentals of multi-omics integration strategies.

- **Practical Skills**:
  - Demonstrate common tools in R (iCluster, MOFA)

- **Real-World Application**:
  - Application in Chronic Lymphocytic Leukemia (CLL)

What You Will Learn

- **Core Competencies**:
  - Data preprocessing: Normalization, batch correction.
  - Integration techniques: Clustering, dimensionality reduction.

- **Common Pitfalls**:
  - High dimensionality, heterogeneity, batch effects, missing data.

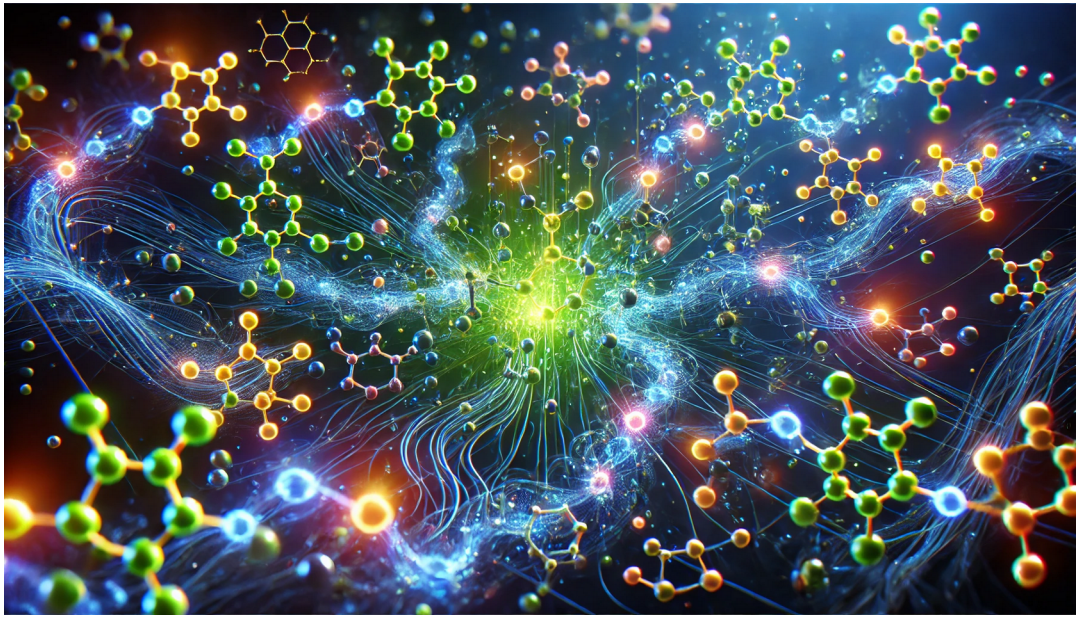
- **Implementation**:
  - R-based tools (iCluster, MOFA)
  - Real dataset (CLL)

Summary

- Multi-omics integration provides a holistic view of biological systems.
- Challenges exist, but computational tools help overcome them.
- Next section: Data Preprocessing & Quality Control.

Outline

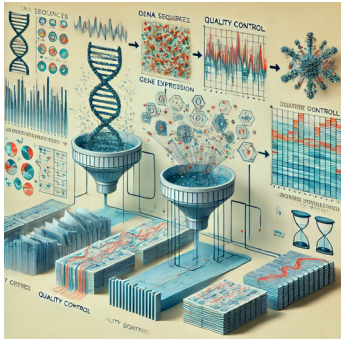
- Introduction
- Preprocessing
- Methods
- Methods Focus
- Case Study
- Summary/Wrap-up/Questions



Importance of Preprocessing

- Raw multi-omics data contains noise, batch effects, and missing values

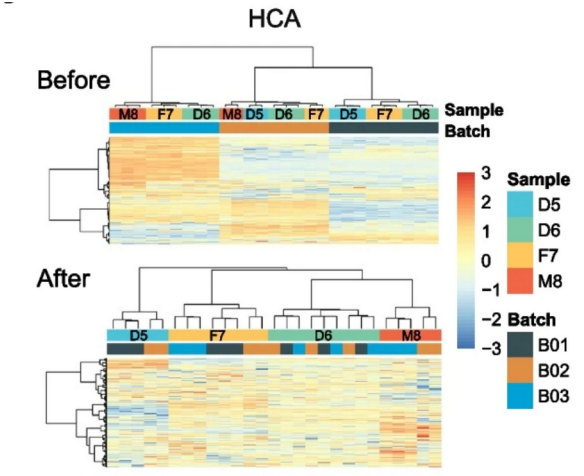
- Proper preprocessing ensures:
  - Reliable biological interpretation
  - Reproducibility across studies



- **Poor preprocessing can lead to false discoveries or missing true discoveries!**

Reproducibility & Reliability

- The reproducibility crisis: Variability in results due to poor quality control
  - **~70% of researchers were unable to reproduce the findings of other scientists and ~60% of researchers could not reproduce their own findings in biological fields (Baker M. Nature News Feature, May 25, 2016)**
- Ensuring reliability:
  - Using robust preprocessing pipelines is one part
- Example: Uncorrected batch effects causing clustering by lab instead of biology, leading to false association



Yu Y, Mai Y, Zheng Y, Shi L. Assessing and mitigating batch effects in large-scale omics studies. Genome Biol. 2024 Oct 3;25(1):254.

Preprocessing Order of Operations

1. Filtering

- remove poor quality/low information features; internal controls

2. Normalization

- Normalizing upfront prevents large technical variations from overshadowing true signal

3. Batch Effect Correction

- remove non-biological variation between experimental batches or platforms

4. Missing Data Imputation

- Imputing after batch correction avoids introducing false structure during batch adjustment

5. Scaling

- Last step to get data analysis ready.

Preprocessing Order of Operations

1. Filtering

- remove poor quality/low information features; internal controls

2. Normalization

- Normalizing upfront prevents large technical variations from overshadowing true signal

3. Batch Effect Correction

- remove non-biological variation between experimental batches or platforms

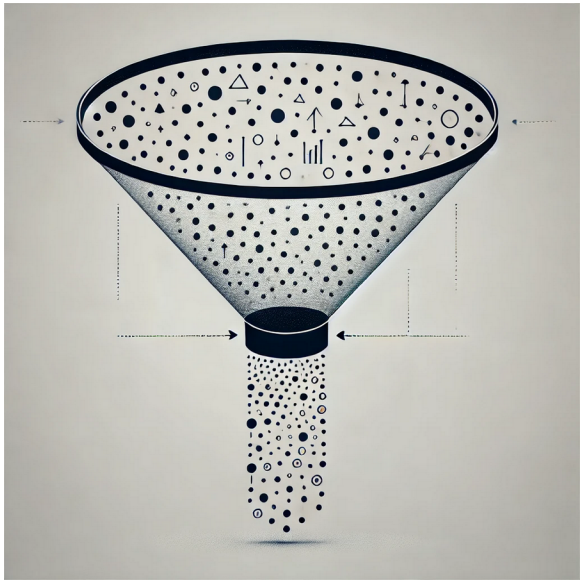
4. Missing Data Imputation

- Imputing after batch correction avoids introducing false structure during batch adjustment

5. Scaling

- Last step to get data analysis ready.

Filtering



- Removes noise from feature sets
- Options
  - **Quality** (detection p-value, missingness, etc.)
  - **Abundance**
  - **Variability**
  - **Association** (e.g., LASSO [glmnet] or mRMR [mRMRe])
  - **Function based** (gene ontology, pathway)

minimal-redundancy-maximal-relevance criterion (mRMR); Hanchuan Peng, Fuhui Long and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1226-1238, Aug. 2005,

Preprocessing Order of Operations

1. **Filtering**
   - remove poor quality/low information features; internal controls

2. **Normalization**
   - Normalizing upfront prevents large technical variations from overshadowing true signal

3. **Batch Effect Correction**
   - remove non-biological variation between experimental batches or platforms

4. **Missing Data Imputation**
   - Imputing after batch correction avoids introducing false structure during batch adjustment

5. **Scaling**
   - Last step to get data analysis ready.

Normalization Techniques

- Objective: Ensure comparability across features & samples.

- Methods:

  - **Z-Score Normalization**: Standardizes data (mean = 0, SD = 1)
  - **Quantile Normalization:** Matches distributions across samples
  - **CPM/RPKM/FPKM:** RNA-seq normalization methods
  - **TMM –** trimmed mean of M-values (edgeR)

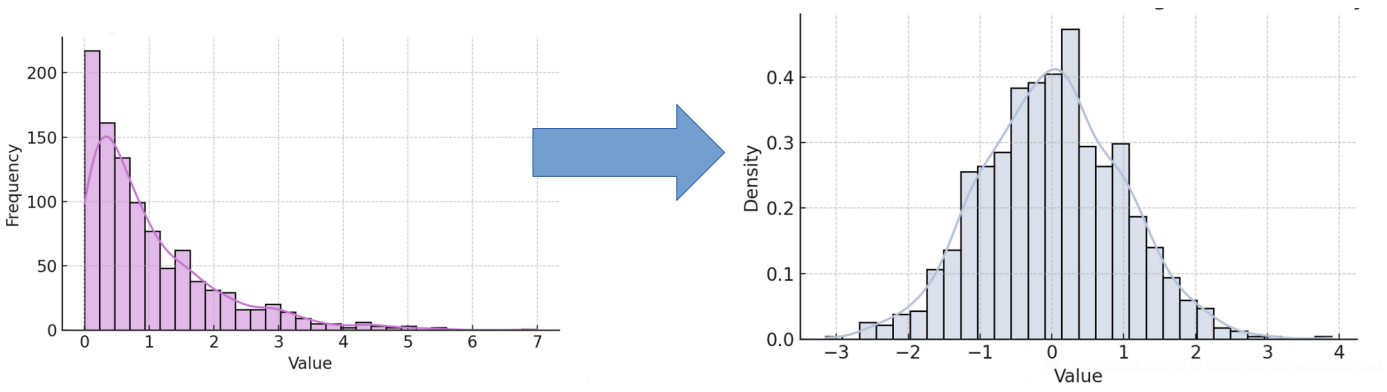
Z-Score Normalization

Cheadle C, Vawter MP, Freed WJ, Becker KG. Analysis of microarray data using Z score transformation. J Mol Diagn. 2003 May;5(2):73-81.

- All features have mean = 0 and sd = 1

$$Z = \frac{X - \mu}{\sigma}$$

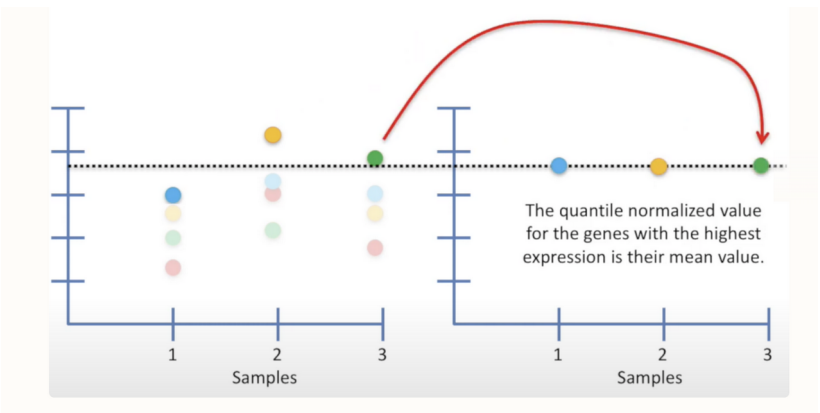
Gene A



Quantile Normalization

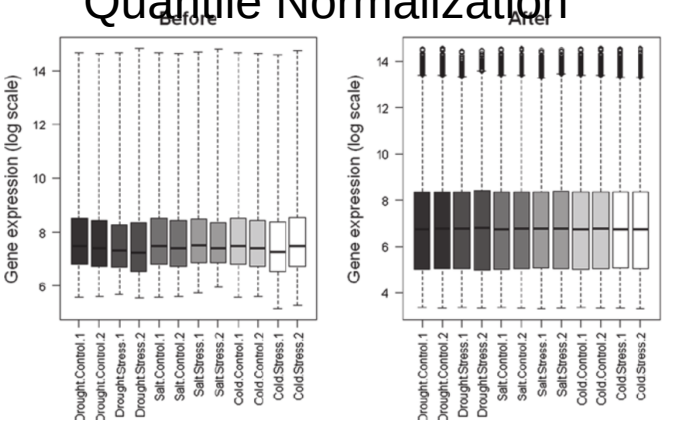
B.M. Bolstad, R.A Irizarry, M. Åstrand, T.P. Speed, A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, Bioinformatics, Volume 19, Issue 2, January 2003, Pages 185–193

- Samples have same distribution across genes
  - Rank genes by expression within sample
  - Map to quantiles in chosen distribution



https://notes.asterhu.com/notes/quantile_normalization

Quantile Normalization



Mohapatra, Saroj & Krishnan, Arjun. (2011). Microarray Data Analysis. Methods in molecular biology (Clifton, N.J.). 678. 27-43. 10.1007/978-1-60761-682-5_3.

CPM/TPM/RPKM

- CPM – counts per million
  - Per gene i:

$$\mathrm{CPM}_i = \frac{\mathrm{counts}_i}{\sum_{j=1}^{N} \mathrm{counts}_j} \times 10^6$$

Where:

- CPM_i is the normalized count (CPM) for gene i.
- counts_i is the raw read count of gene i.
- $\sum_{j=1}^{N} \mathrm{counts}_j$ is the sum of raw counts for all N genes in the sample.
- 10^6 is the scaling factor to express counts per million.

- RPKM/FPKM - reads/fragments per kilobase of exon per million reads mapped
  - Per gene i:

$$RPKM_i \text{ or } FPKM_i = \frac{q_i}{\frac{l_i}{10^3} * \frac{\sum_j q_j}{10^6}} = \frac{q_i}{l_i * \sum_j q_j} * 10^9$$

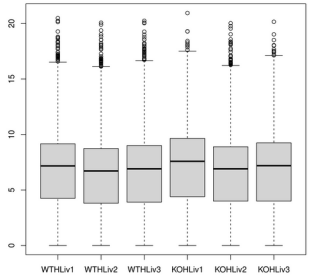
where q_i are raw read or fragment counts, l_i is feature (i.e., gene or transcript) length, and $\sum_j q_j$ corresponds to the total number of mapped reads or fragments.

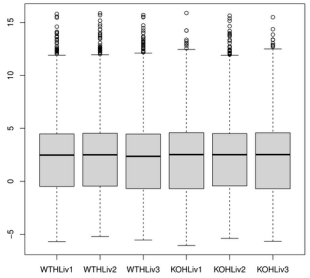
- TPM - transcript per million
  - Per gene i:

$$TPM_i = \frac{q_i / l_i}{\sum_j (q_j / l_j)} * 10^6$$

where q_i denotes reads mapped to transcript, l_i is the transcript length, and $\sum_j (q_j / l_j)$ corresponds to the sum of mapped reads to transcript normalized by transcript length.

Zhao Y, Li MC, Konaté MM, Chen L, Das B, Karlovich C, Williams PM, Evrard YA, Doroshow JH, McShane LM. TPM, FPKM, or Normalized Counts? A Comparative Study of Quantification Measures for the Analysis of RNA-seq Data from the NCI Patient-Derived Models Repository. J Transl Med. 2021 Jun 22;19(1):269.





TMM

- Trimmed mean of M-values (edgeR)
  1. Calculate M (FC) and A (ave expn) values
  2. Trim genes with extreme M and A values
     - Remove genes that likely have a real association
  3. Weighted mean of remaining M-values (TMM factor)
     - Each gene's contribution is weighted inversely by the variance of its log fold change
  4. Use TMM to adjust effective library sizes

edgeR package: y <- calcNormFactors(y, method="TMM")

- Y_{gk} and Y_{gr} are raw counts for gene g in sample k and reference sample r.
- N_k and N_r are total library sizes for samples k and r.

- **M-value (log fold change)**:

$$M_g = \log_2 \left(\frac{Y_{gk}/N_k}{Y_{gr}/N_r} \right)$$

- **Weighted TMM Normalization Factor**:

$$\log_2(\mathrm{TMM}_k^r) = \frac{\sum_{g \in G^*} w_{gk}^r M_{gk}^r}{\sum_{g \in G^*} w_{gk}^r}$$

- **Gene weights (inverse variance)**:

$$w_{gk}^r = \frac{N_k - Y_{gk}}{N_k Y_{gk}} + \frac{N_r - Y_{gr}}{N_r Y_{gr}}$$

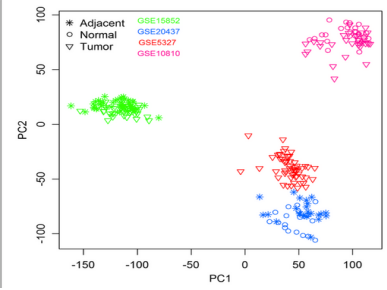
- **Adjusted (effective) library sizes**:

$$N_k^{eff} = N_k \times \sqrt{\mathrm{TMM}_k^r} \quad ; \quad N_r^{eff} = \frac{N_r}{\sqrt{\mathrm{TMM}_k^r}}$$

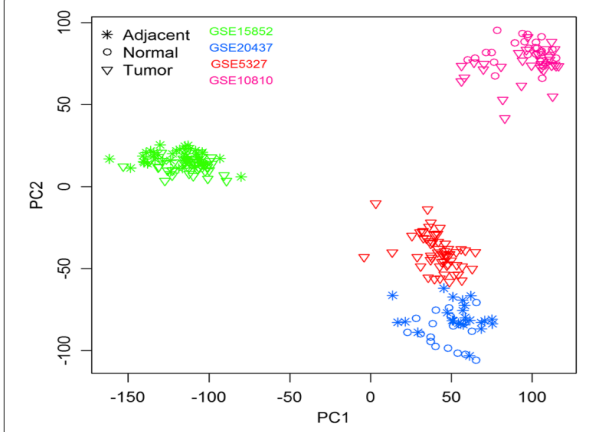
Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biol 2010;11(3):R25.

Visualization Techniques

- Purpose: Visualization to inspect data for batch effects, other artifacts
- Typically performed after normalization
- Methods:
  - **PCA**: Linear transformation maximizing variance
  - **t-SNE** & **UMAP**: Non-linear embedding techniques for visualization



PCA



- Principal Component Analysis (prcomp())
- Transforms high-dimensional omics data into a smaller set of uncorrelated components while preserving the most important variation; linear method
- Identifies new axes that capture the most variance in the data
- Reduces noise and redundancy while preserving major biological patterns

t-SNE & UMAP

- t-SNE (t-Distributed Stochastic Neighbor Embedding) and UMAP (Uniform Manifold Approximation and Projection) (Rtsne, umap packages)

- unsupervised non-linear dimensionality reduction techniques used for visualizing high-dimensional omics data

- t-SNE preserves local similarities between points but can distort global structure

- UMAP preserves both local and some global structures, making it more scalable and better suited for clustering

PCA, t-SNE, UMAP

Method	Best For	Advantages	Limitations
PCA (Principal Component Analysis)	Capturing linear structure, feature reduction, preprocessing	Fast, interpretable, maintains global structure	Only captures **linear** relationships, may miss complex patterns
t-SNE (t-Distributed Stochastic Neighbor Embedding)	Visualizing clusters in high-dimensional data	Preserves **local structure**, useful for exploring subpopulations	Computationally expensive, distorts global structure, non-deterministic
UMAP (Uniform Manifold Approximation and Projection)	Clustering & visualization with large datasets	**Faster & more scalable** than t-SNE, preserves local & some global structure	Less interpretable than PCA, requires parameter tuning

Preprocessing Order of Operations

1. **Filtering**
   - remove poor quality/low information features; internal controls

2. **Normalization**
   - Normalizing upfront prevents large technical variations from overshadowing true signal

3. **Batch Effect Correction**
   - remove non-biological variation between experimental batches or platforms

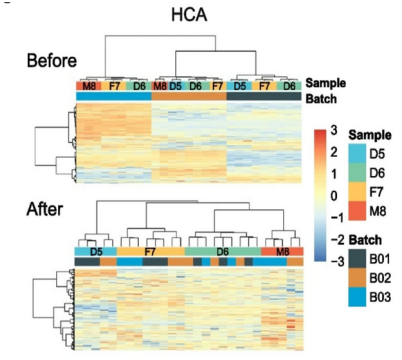
4. **Missing Data Imputation**
   - Imputing after batch correction avoids introducing false structure during batch adjustment

5. **Scaling**
   - Last step to get data analysis ready.

Batch Effect Correction

- What are batch effects?
  - Systematic differences due to lab, operator, or sequencing run.

- Correction methods:
  - **ComBat** (Empirical Bayes adjustment).
  - **sva** (Surrogate Variable Analysis)



Yu Y, Mai Y, Zheng Y, Shi L. Assessing and mitigating batch effects in large-scale omics studies. Genome Biol. 2024 Oct 3;25(1):254.

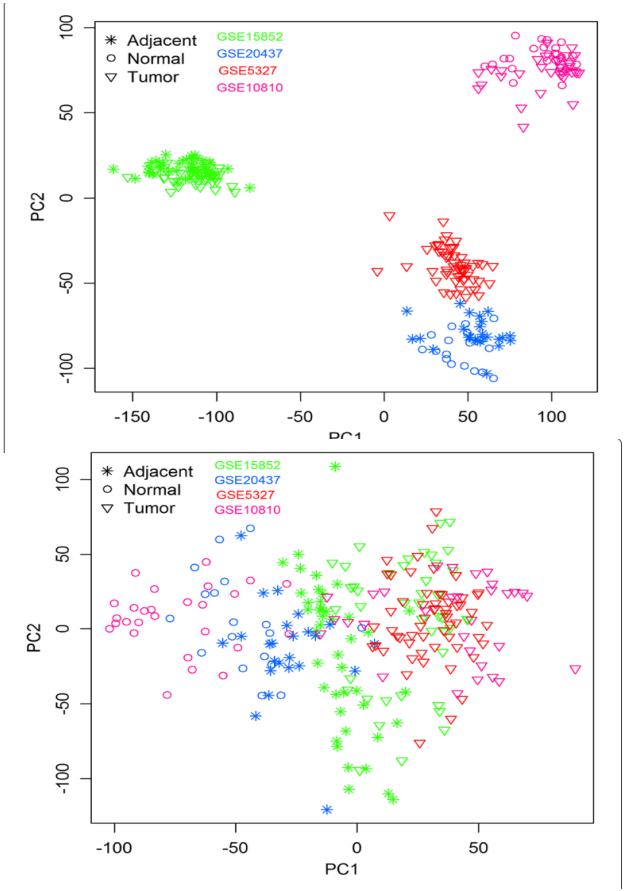
ComBat

W. Evan Johnson, Cheng Li, Ariel Rabinovic, Adjusting batch effects in microarray expression data using empirical Bayes methods, Biostatistics, Volume 8, Issue 1, January 2007, Pages 118–127

- Empirical Bayes adjustment (ComBat function; sva package)
  - Uses info across all genes to estimate batch effects
  - Shrinks the correction for each gene toward the overall average

$$Y_{ijg} = \alpha_g + X_j\beta_g + \gamma_{ig} + \delta_{ig}\,\varepsilon_{ijg}$$

Variable	Meaning
Y_{ijg}	Observed expression level of **gene** g in **sample** j from **batch** i.
α_g	Baseline expression level of **gene** g across all samples.
$X_j\beta_g$	Biological effects (e.g., disease vs. control, treatment effects). This ensures true biological differences are **preserved** in the correction.
γ_{ig}	**Batch effect**: How much the expression of **gene** g is shifted **up or down** in batch i.
δ_{ig}	**Batch-specific variance**: Adjusts for differences in **variability (spread)** of gene expression in batch i.
ε_{ijg}	Random noise: Unexplained variation after accounting for batch and biological effects.



Before and after ComBat batch correction.

Thillaiyampalam, Gayathri & Liberante, Fabio & Murray, Liam & Cardwell, Chris & Mills, Ken & Zhang, Shu-Dong. (2017). An integrated meta-analysis approach to identifying medications with potential to alter breast cancer risk through connectivity mapping. BMC Bioinformatics. 18. 10.

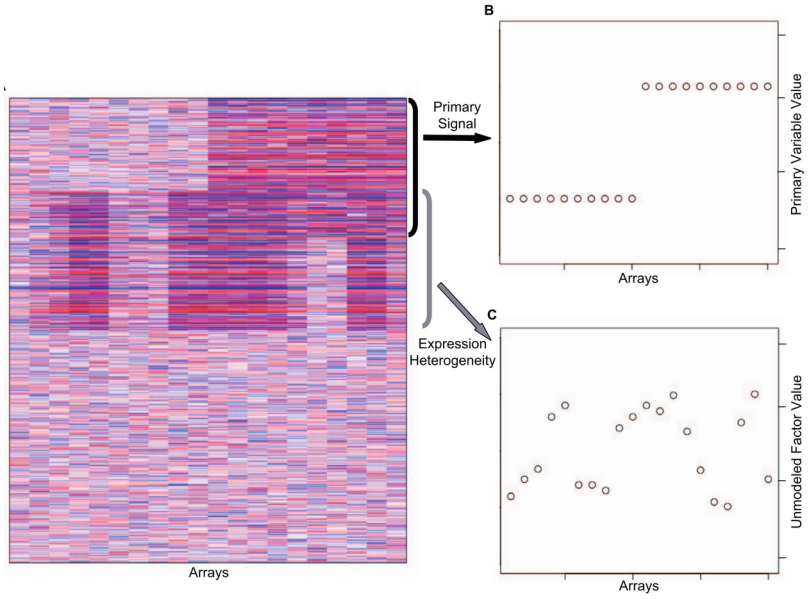
SVA

Leek JT, Storey JD. Capturing heterogeneity in gene expression studies by surrogate variable analysis. PLoS Genet. 2007 Sep;3(9):1724-35.

- Surrogate Variable Analysis (sva package)
  - Fit primary model
  - Singular value decomposition on residuals
  - Construct surrogate variables
  - Use as covariables in model

$$Y_{ij} = \mu_i + f_i(\text{primary variables}) + \sum_{k=1}^{K} \gamma_{ik} S_{kj} + \varepsilon_{ij}$$

Variable	Meaning
Y_{ij}	Expression of gene i in sample j.
μ_i	Baseline expression level of gene i.
$f_i(\text{primary variables})$	Effects of known primary variables (e.g., treatment).
S_{kj}	Surrogate variable k (hidden variation) in sample j.
γ_{ik}	Effect of surrogate variable k on gene i.
ε_{ij}	Gene-specific random error.



Preprocessing Order of Operations

1. **Filtering**
   - remove poor quality/low information features; internal controls

2. **Normalization**
   - Normalizing upfront prevents large technical variations from overshadowing true signal

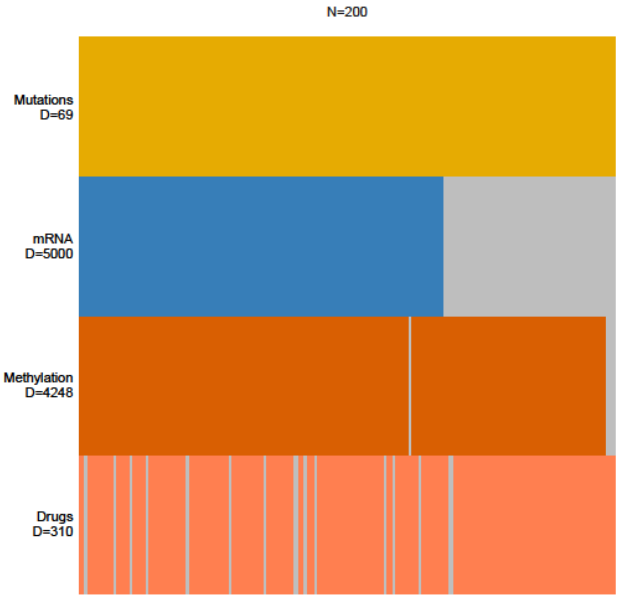
3. **Batch Effect Correction**
   - remove non-biological variation between experimental batches or platforms

4. **Missing Data Imputation**
   - Imputing after batch correction avoids introducing false structure during batch adjustment

5. **Scaling**
   - Last step to get data analysis ready.

Handling Missing Data

- **Causes:**
  - Not all omics layers are measured for all samples
  - Technical failures
- **Imputation strategies:**
  - Simple:
    - Mean/median imputation
    - Complete Case
  - Advanced:
    - k-Nearest Neighbors (kNN) **impute::impute.knn()**
    - Matrix completion (SVD, softImpute) **softImpute::softImpute()**
    - MICE **mice::mice()**
    - Random forest-imputation **missForest::missForest()**



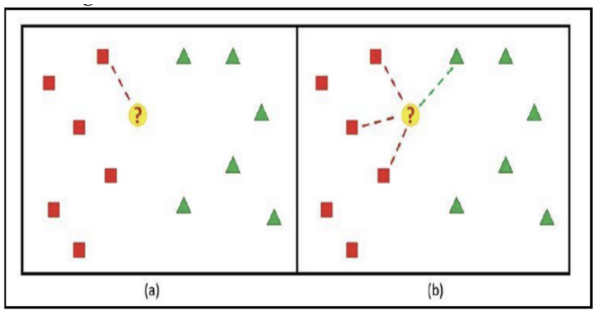
k-Nearest Neighbors (kNN)

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, Russ B. Altman, Missing value estimation methods for DNA microarrays , Bioinformatics, Volume 17, Issue 6, June 2001, Pages 520–525

- Estimates missing gene expression values based on similar genes
- Assumes genes with similar expression patterns across experiments behave similarly.
  1. Identify K most similar genes (nearest neighbors) based on Euclidean distance
  2. Calculate a weighted average of the observed expression values from these neighbors
  3. Use this weighted average to replace the missing value

$$\text{Imputed Value}_A = \frac{\sum_{i=1}^{K} (w_i \times \text{Value}_i)}{\sum_{i=1}^{K} w_i}$$

- Value_i = Expression value from neighbor gene i.
- w_i = Weight based on similarity (usually inverse Euclidean distance).



Imandoust, Sadegh Bafandeh and Mohammad Bolandraftar. “Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background.” (2013).

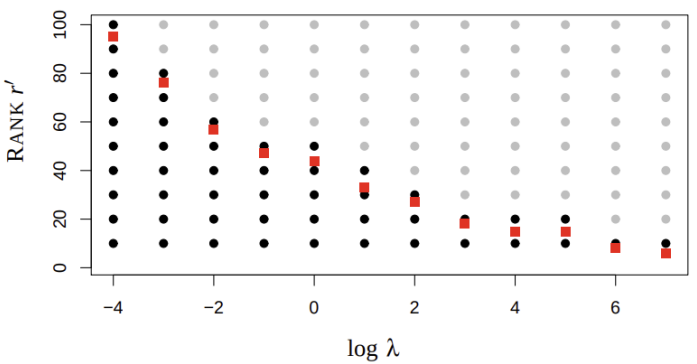
Impute::impute.knn()

Matrix completion

Rahul Mazumder, Trevor Hastie and Rob Tibshirani (2010) Spectral Regularization Algorithms for Learning Large Incomplete Matrices, Journal of Machine Learning Research 11 (2010) 2287-2322

- Singular value decomposition (SVD), softImpute

1. Initialize missing entries (e.g., with zeros or column means)
2. Compute Singular value decomposition (SVD) of the completed matrix
3. Perform soft-thresholding on singular values (shrink towards zero) $\text{soft-threshold}(d_i, \lambda) = (d_i - \lambda)_+$
4. Update missing entries with reconstructed values from thresholded SVD
5. Iterate until convergence (values stabilize)



MICE

van Buuren S, Groothuis-Oudshoorn K (2011). “mice: Multivariate Imputation by Chained Equations in R.” Journal of Statistical Software, 45(3), 1-67

- MICE - multiple imputation by chained equations; mice::mice()

1 Initialization: Replace missing values with initial estimates (e.g., mean, random values).

2 Iterative Process: For each variable with missing data:

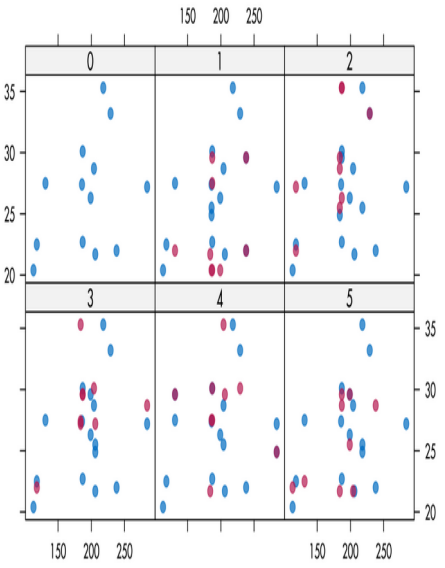
Treat this variable as the outcome, and the remaining variables as predictors.

Build a predictive model (e.g., regression model) using only cases with observed values.

Replace missing values by drawing predictions from this model.

Cycle through all incomplete variables until the process converges (usually 10–20 iterations).

3 Create Multiple Datasets: Repeat this entire iterative procedure multiple times, resulting in multiple complete datasets. Differences across datasets reflect uncertainty about the true missing values.



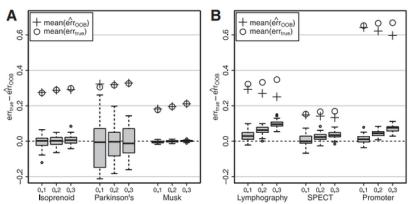
Random forest-imputation

Stekhoven DJ, Buehlmann P (2012). “MissForest - non-parametric missing value imputation for mixed-type data.” Bioinformatics, 28(1), 112–118.

- Non-parametric Imputation Using Random Forests (missForest::missForest())
- MissForest handles complex, non-linear relationships and mixed data types without strong assumptions.

1. Initial guess: Start by filling in missing values with simple estimates (e.g., mean or mode).
2. Iterative Imputation: For each variable with missing values:
   - Train a Random Forest model using observations without missing values in this variable.
   - Use the trained Random Forest to predict missing values.

   Update your dataset with predicted values.
3. Repeat until convergence: Keep repeating step 2, cycling through variables, until the imputed values stabilize.



Preprocessing Order of Operations

1. Filtering

- remove poor quality/low information features; internal controls

2. Normalization

- Normalizing upfront prevents large technical variations from overshadowing true signal

3. Batch Effect Correction

- remove non-biological variation between experimental batches or platforms

4. Missing Data Imputation

- Imputing after batch correction avoids introducing false structure during batch adjustment

5. Scaling

- Last step to get data analysis ready

Scaling in Multi-Omics Data

- Different omics layers have different data distributions:
  - RNA-seq: Count data (often needs log transformation).
  - Metabolomics: Continuous data with large dynamic range.
  - Epigenomics: Fractional methylation values (0-1 range; M-Value transformation).

- Without proper scaling, some features dominate the analysis, leading to biased results.
- Can be part of normalization

Scaling

Method	Formula	Pros	Cons	When to Use
Autoscaling	$\frac{X-\mu}{\sigma}$	Equal feature importance	Sensitive to outliers	PCA, clustering, classification models
Pareto scaling	$\frac{X-\mu}{\sqrt{\sigma}}$	Reduces dominance of large features	Still sensitive to noise	Metabolomics, when large variance features are biologically relevant but shouldn't overshadow
Vast scaling	$\frac{X-\mu}{\sigma}\frac{\mu}{\sigma}$	Highlights small variations	Can exaggerate minor fluctuations	Detecting subtle signals or rare biomarkers
Min-Max	$\frac{X-X_{min}}{X_{max}-X_{min}}$	Simple; bounded range [0,1]	Highly sensitive to outliers	Neural networks, visualizations, applications needing data in a fixed range
Centering	$X-\mu$	Preserves variance structure	Doesn't equalize scales	When absolute differences are meaningful; variance informative (gene expression studies)
Level scaling	$\frac{X-\mu}{\mu}$	Emphasizes relative (%) changes	Sensitive to small means	Comparing relative changes, percentage differences
Log transform	$\log(X)$	Reduces skewness; stabilizes variance	Undefined for zeros	RNA-seq counts, metabolomics, proteomics, microarray intensities (skewed data)
Power transform	$\frac{X^{\lambda}-1}{\lambda}$ (Box-Cox)	Flexible; stabilizes variance	Less intuitive; choice of parameter needed	Non-normal data with heteroscedasticity (changing variance), flexible normalization

Preprocessing Order of Operations

1. Filtering

- remove poor quality/low information features; internal controls

2. Normalization

- Normalizing upfront prevents large technical variations from overshadowing true signal

2. Batch Effect Correction

- remove non-biological variation between experimental batches or platforms

3. Missing Data Imputation

- Imputing after batch correction avoids introducing false structure during batch adjustment

4. Scaling

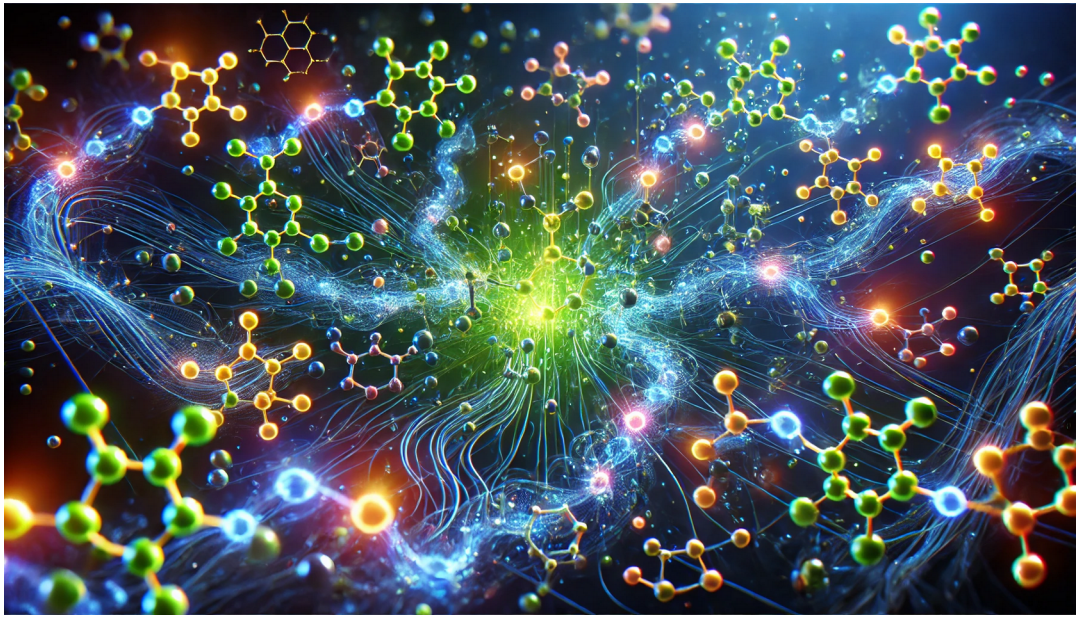
- Last step to get data analysis ready.

Summary

- Preprocessing is essential for accurate integration & downstream analysis.
- Normalization, batch correction, and feature selection reduce artifacts.
- Next section: Approaches to Multi-Omics Integration.

Outline

- Introduction
- Preprocessing
- Methods
- Methods Focus
- Case Study
- Summary/Wrap-up/Questions



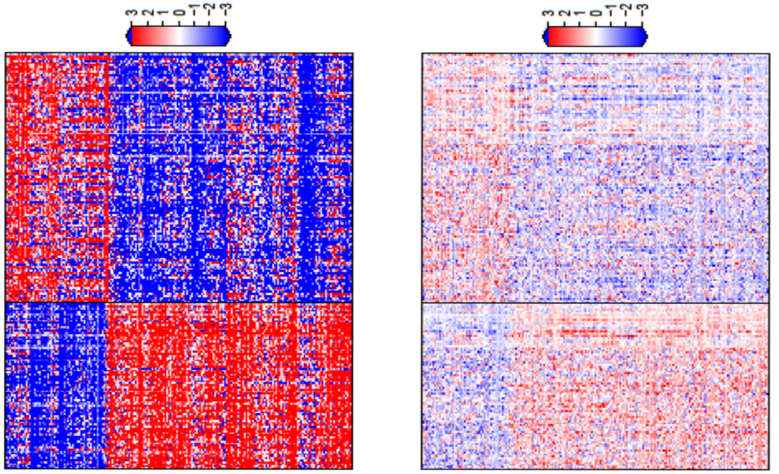
Overview of Multi-Omics Integration Methods

Vahabi N, Michailidis G. Unsupervised Multi-Omics Data Integration Methods: A Comprehensive Review. Front Genet. 2022 Mar 22;13:854752

- Goal: Combine multiple omics layers for deeper biological insights
  - **Unsupervised** - Analyze multi-omics data without predefined labels, identifying hidden patterns, clusters, or associations among samples, commonly using clustering, dimensionality reduction, or network-based approaches for exploratory analysis
  - Supervised

Unsupervised Methods

- **Vahabi et al (2022)** classifies methods into three “comprehensive” categories and three “macro” categories.
- **Comprehensive**
  - Regression/Association-based methods
  - Clustering-based methods
  - Network-based methods
- **Macro**
  - Multi-step and Sequential Analysis (MS-SA)
  - Data-ensemble (DatE)
  - Model-ensemble (ModE)
- **Note: Each method belongs to both a comprehensive and macro category**

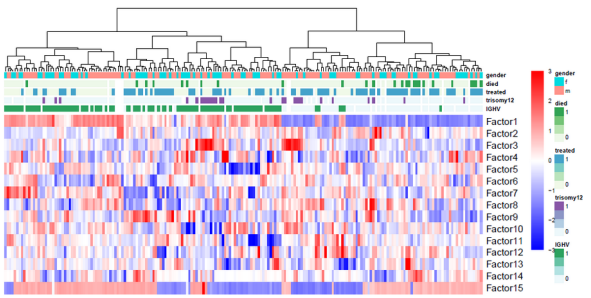


Macro Categories

- **Multi-step and Sequential Analysis (MS-SA)**
  - Applies stepwise analysis where one omics layer is used to guide the selection or integration of features in another omics layer
  - Often used in biomarker discovery by progressively filtering relevant molecular features
  - Examples: CNAmet, MEMo, iPAC
- **Data-ensemble (DatE)**
  - Concatenates multiple omics data into a single feature matrix before applying unsupervised learning, treating all omics as a unified dataset
  - Allows comprehensive multi-omics integration but may suffer from high dimensionality
  - Examples: **MOFA**, MCIA, FA (Factor Analysis)
- **Model-ensemble (ModE)**
  - Analyzes each omics type separately and then fuses results, often using consensus clustering or network-based methods
  - Provides robustness by combining multiple views of biological data while maintaining individual omics-specific characteristics
  - Examples:**iCluster**, SNF, Bayesian Consensus Clustering (BCC)

Comprehensive Categories

- **Regression/Association-based methods**
  - **Identify statistical associations between different omics layers using correlation, regression, or latent factor models to uncover biologically relevant relationships**
- **Clustering-based methods**
- **Network-based methods**



Regression/Association-Based Methods

Sequential Analysis Methods - integrate multi-omics data through a stepwise process, where one omics layer informs the analysis of another, often used to prioritize features or refine associations progressively

- **CNAMet**: Integration of CNV and methylation with gene expression (MS-SA)
- **MEMo (Mutual Exclusivity Modules)**: Identifies mutually exclusive genomic events (MS-SA)
- **IPAC (in-trans Process Associated and cis-Correlated)**: Identifies genes regulated by copy number variation (MS-SA)

Canonical Correlation (CCA)-based Methods - identify linear relationships between multiple omics datasets by finding pairs of components (one from each dataset) that are maximally correlated, capturing shared variation across omics layers

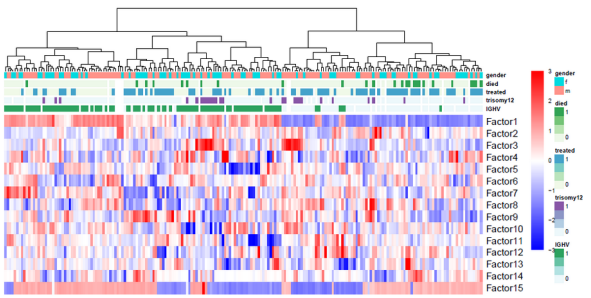
- **Sparse MCCA (Sparse Multiple Canonical Correlation Analysis)**: Sparse components via regularization (DatE)
- **MCIA (Multiple Co-Inertia Analysis)**: Co-analysis across multiple omics datasets (DatE)

Factor Analysis-based Methods - decompose multi-omics datasets into a lower-dimensional representation by identifying latent factors that explain the major sources of variation, helping to uncover hidden biological structures

- **MOFA (Multi-Omics Factor Analysis)**: Captures latent factors and sources of variation (DatE)
- **Bayesian Relational Learning (BayRel)**: Bayesian inference for integrated datasets (DatE)

Comprehensive Categories

- **Regression/Association-based methods**
- **Clustering-based methods**
  - Group samples into biologically meaningful subtypes using similarity measures, matrix factorization, or probabilistic clustering
- **Network-based methods**



Clustering-Based Methods

Kernel-Based Methods - transform multi-omics data into similarity kernels, capturing complex nonlinear relationships between samples and enabling more robust clustering or integration in a high-dimensional space

- **SNF (Similarity Network Fusion)**: Fuses multiple omics data similarity networks (ModE)
- **ANF (Affinity Network Fusion)**: Enhances SNF by capturing local data structures (ModE)
- **mixKernel**: Integrates kernels from different omics modalities (ModE)

Matrix Factorization-Based Methods - decompose multi-omics datasets into lower-dimensional latent components, separating shared and dataset-specific signals to improve clustering

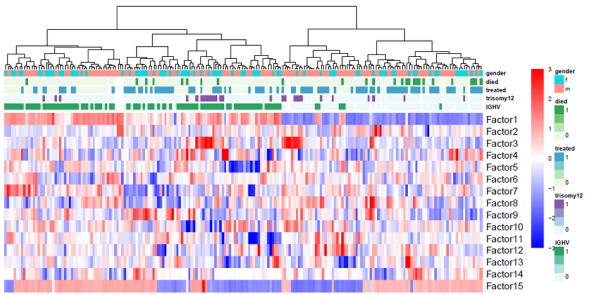
- **iCluster/iClusterPlus**: Latent variable modeling for integrated clustering (ModE)
- **Joint Non-negative Matrix Factorization (jNMF)**: Joint analysis across omics modalities (ModE)
- **integrative NMF (iNMF)**: Integrates data by non-negative factorization (ModE)

Bayesian Clustering Methods - Probabilistic methods that model uncertainty in clustering assignments by using prior distributions

- **PARADIGM**: Bayesian approach using known pathways for integration (ModE)
- **MDI (Multiple Dataset Integration)**: Bayesian clustering across multiple omics datasets (ModE)
- **Bayesian Consensus Clustering (BCC)**: Consensus clustering using Bayesian methods (ModE)

Comprehensive Categories

- **Regression/Association-based methods**
- **Clustering-based methods**



- **Network-based methods**
  - **Construct biological networks to detect functional modules and disease-associated pathways by leveraging prior knowledge and statistical associations**

Network-based Methods

Matrix Factorization-Based Networks - apply matrix decomposition techniques to reconstruct biological networks from multi-omics data, identifying shared and dataset-specific patterns that capture functional relationships

- **Network-Based Stratification (NBS)**: Stratifies patient data into clinically relevant subtypes (ModE)
- **DisoFun**: Discovers disease-related functions from multi-omics data (ModE)

Bayesian Networks - Probabilistic graphical models that represent dependencies between molecular entities in a network structure

- **CONEXIC**: Bayesian network modeling integrating CNV and gene expression (ModE)
- **PARADIGM** (also listed under Bayesian Clustering): Uses Bayesian networks on known pathways (ModE)

Network Propagation-Based Networks - spread information across a biological network using diffusion or propagation algorithms

- **HotNet2**: Identifies significantly altered subnetworks via heat diffusion (ModE)
- **SNF** (also listed under Kernel-Based): Similarity-based fusion for network propagation (ModE)
- **TieDIE**: Integrates data using network diffusion models (ModE)

Correlation-Based and Other Networks - construct networks by quantifying statistical dependencies (e.g., co-expression, mutual information) between molecular features

- **Weighted Gene Co-expression Network Analysis (WGCNA)**: Co-expression network analysis for identifying gene modules (DatE)
- **Graphical Gaussian Models (GGM)**: Conditional independence network analysis (ModE)

Overview of Multi-Omics Integration Methods

- Goal: Combine multiple omics layers for deeper biological insights
  - Unsupervised
  - **Supervised -** Utilize labeled data (e.g., known disease subtypes, clinical outcomes) to train models that predict outcomes or classify samples, often employing machine learning or regression-based approaches for biomarker discovery and patient stratification

Supervised Integration: Predictive Modeling

- Goal: Use multi-omics data to predict outcomes (e.g., disease status, survival).
- Common methods:
  - **LASSO Regression**: Shrinks coefficients to select important features.
  - **Random Forests**: Ensemble-based decision tree approach.
  - **Support Vector Machines (SVM)**: Finds optimal decision boundaries.
  - **Deep Learning**: Autoencoders & CNNs for feature extraction.

LASSO Regression

- **Least Absolute Shrinkage and Selection Operator (LASSO) regression (glmnet)**
  - Linear regression method that performs both variable selection and regularization to enhance predictive performance and interpretability
  - Prevents overfitting by enforcing sparsity (penalization), making models more generalizable (less sensitive to small variations in data)
  - model reduces the magnitude of less important coefficients, shrinking some to 0

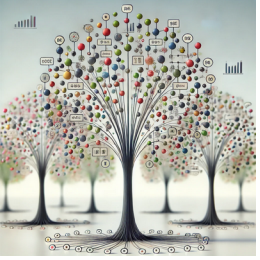
Least squares loss function Regularization penalty

$$\min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

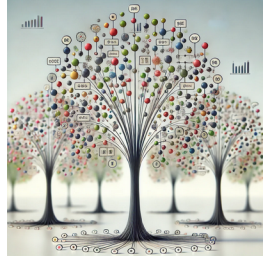
$\min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

Where:

- **y** = response variable (e.g., patient survival, disease subtype)
- **X** = predictor variables (e.g., omics features)
- **β** = regression coefficients
- **λ** = tuning parameter controlling the strength of regularization



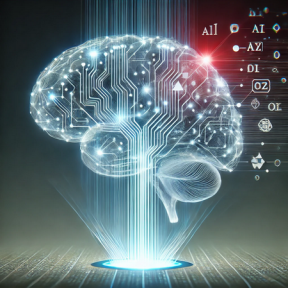
Random Forests



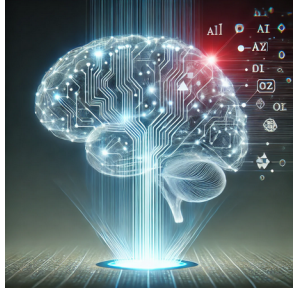
- Ensemble learning method that builds multiple decision trees and combines their predictions (randomForest)
- Uses bootstrap aggregation (bagging) to train multiple decision trees on different subsets of data
- Final prediction is determined by averaging predictions (regression) or majority voting (classification)

Support Vector Machines (SVM)

- Finds optimal hyperplane to separate classes in high-dimensional space (e1071::svm())
- Identifies the decision boundary that maximizes the margin between different classes, improving classification accuracy
- Works well with small sample sizes



Deep Learning



- Uses multi-layered artificial neural networks to learn complex patterns in large-scale omics data
- Deep learning models automatically extract high-level features from raw omics data through multiple hidden layers
- Results are difficult to interpret; requires large data sets

Practical Considerations: Choosing the Right Method

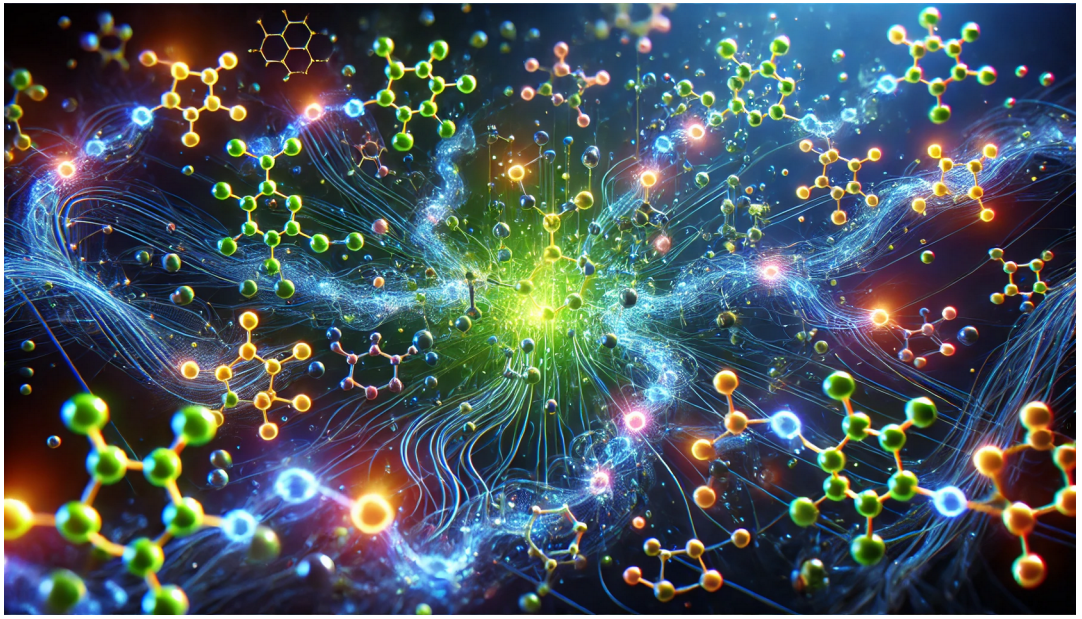
- Factors influencing method selection:
  - Data type & distribution (counts, continuous, categorical).
  - Need for interpretability (e.g., PCA vs. deep learning).
  - Sample size constraints.
  - Computational complexity.

Summary: Key Takeaways

- Integration of multiple omics layers provides deeper biological insights.
- Proper preprocessing (scaling, missing data handling) is crucial.
- Method choice depends on data properties & research objectives.
- Network-based & AI approaches hold promise for future advances in multi-omics integration.

Outline

- Introduction
- Preprocessing
- Methods
- Methods Focus
- Case Study
- Summary/Wrap-up/Questions



MOFA (Multi-Omics Factor Analysis)

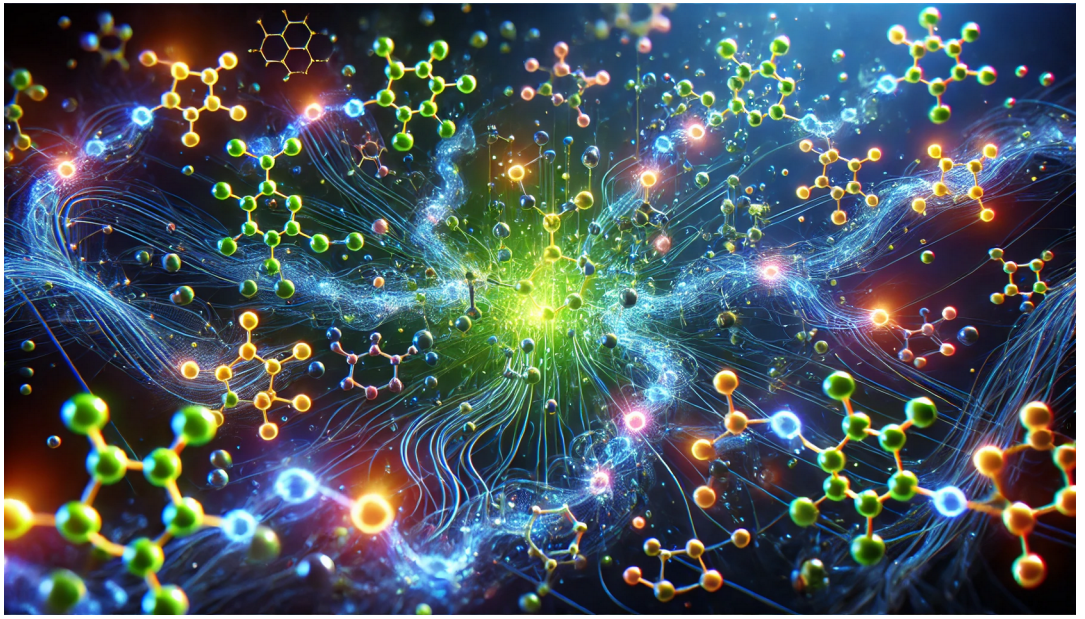
- A Bayesian factor analysis framework for multi-omics integration.
- Key Features:
  - Identifies shared & unique variance across omics layers.
  - Handles missing data
  - Captures latent factors influencing multiple molecular layers
  - Fast
- Applications:
  - Disease subtyping.
  - Biomarker discovery.
  - Multi-modal feature extraction (shared + ‘omics type specific).

iCluster: Integrative Clustering for Multi-Omics

- A latent variable model for joint clustering of multi-omics data.
- Uses a Gaussian mixture model to identify common sample clusters
- Slow
- Two versions:
  - iClusterPlus: Penalized likelihood-based clustering.
  - iClusterBayes: Bayesian framework with priors on clustering structure.
- Applications:
  - Cancer subtyping (e.g., TCGA data).
  - Integrative biomarker identification.

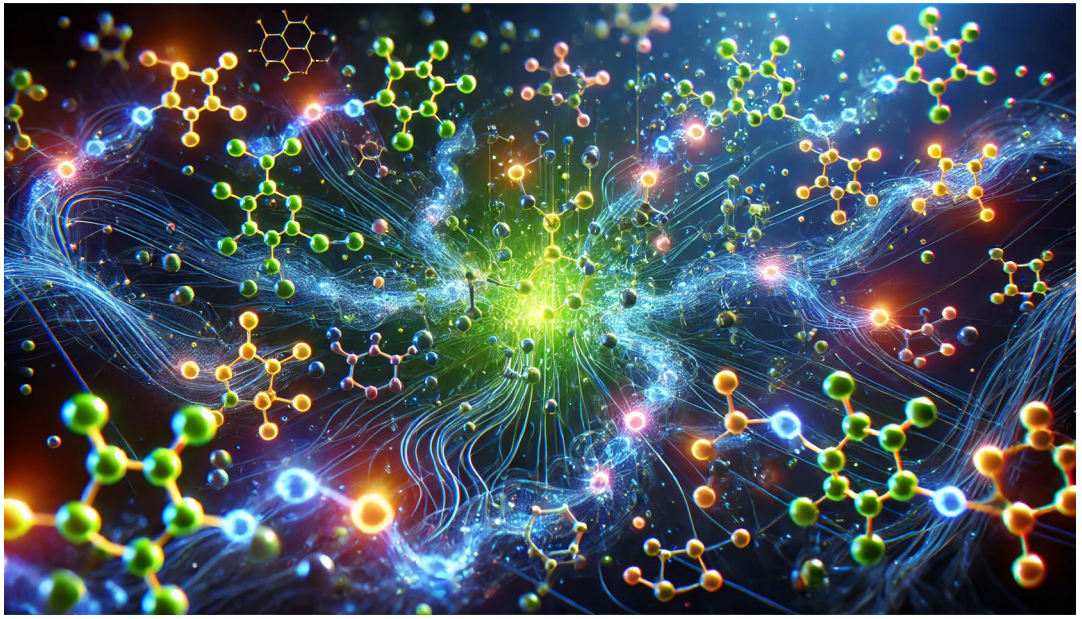
Outline

- Introduction
- Preprocessing
- Methods
- Methods Focus
- Case Study
- Summary/Wrap-up/Questions



Outline

- Introduction
- Preprocessing
- Methods
- Methods Focus
- Case Study
- Summary/Wrap-up/Questions



Key Insights from the Tutorial

- Multi-omics integration can provide a more comprehensive view of biological systems than single-omics
- It enhances biomarker discovery, disease subtyping, and precision medicine.
- Choosing the right integration approach depends on:
  - Data characteristics.
  - Research objectives.
  - Computational constraints.

Strengths of Multi-Omics Integration

- Captures interdependencies across molecular layers.
- Improves patient stratification for targeted therapies.
- Identifies novel biomarkers for early disease detection.
- Enables data-driven drug discovery and repurposing (pathways).
- Bridges the gap between basic research and clinical applications.

Emerging Trends in Multi-Omics

- Single-cell multi-omics:
  - Provides unprecedented resolution of cellular heterogeneity.
- Spatial omics:
  - Integrates molecular and spatial information for tissue-level insights (location).
- AI and machine learning:
  - Deep learning models improve feature extraction and predictive accuracy.

Challenges and Future Directions

- Current challenges:
  - Standardizing multi-omics data formats and integration methods.
  - Handling missing data and batch effects in complex datasets.
  - Scaling computational tools for large multi-omics cohorts.

- Future solutions:
  - Improved statistical models for cross-omics harmonization.
  - Cloud-based and high-performance computing frameworks.
  - Cross-disciplinary collaborations to drive translational research.

Thank you!