Disease subtype discovery

using multi-omics data integration

Introduction

The identification of molecular disease subtypes has emerged as a central objective in cancer research, as it enables a more precise understanding of disease heterogeneity and guides the development of personalized therapeutic strategies.

Multi-omics data, encompassing diverse molecular layers such as gene expression (mRNA), microRNA (miRNA), and protein levels, offers a comprehensive view of the biological processes underlying cancer progression. This holistic approach facilitates the identification of clinically meaningful subtypes that may not be discernible through single-omics analyses.

Despite its potential, the integration and analysis of multi-omics data presents significant challenges. Issues such as missing values, differences in data scale and type, and the selection of appropriate similarity measures complicate the analysis pipeline [1]. The choice of integration strategy can profoundly influence clustering results, where samples are grouped based on molecular similarity. Consequently, robust and efficient integration and clustering methods are essential for meaningful subtype discovery.

In this study, we leverage two approaches for multi-omics data integration: a graph-based algorithm called Similarity Network Fusion (SNF) [2] and a simple averaging of similarity matrices. For clustering, we utilize Partitioning Around Medoids (PAM) [3] and Spectral Clustering [4].

Our goal is to evaluate whether the clusters derived from these multi-omics integration and clustering methods correspond to the molecular subtypes identified by the iCluster integrative model [5] used by *The Cancer Genome Atlas Research Network* [6].

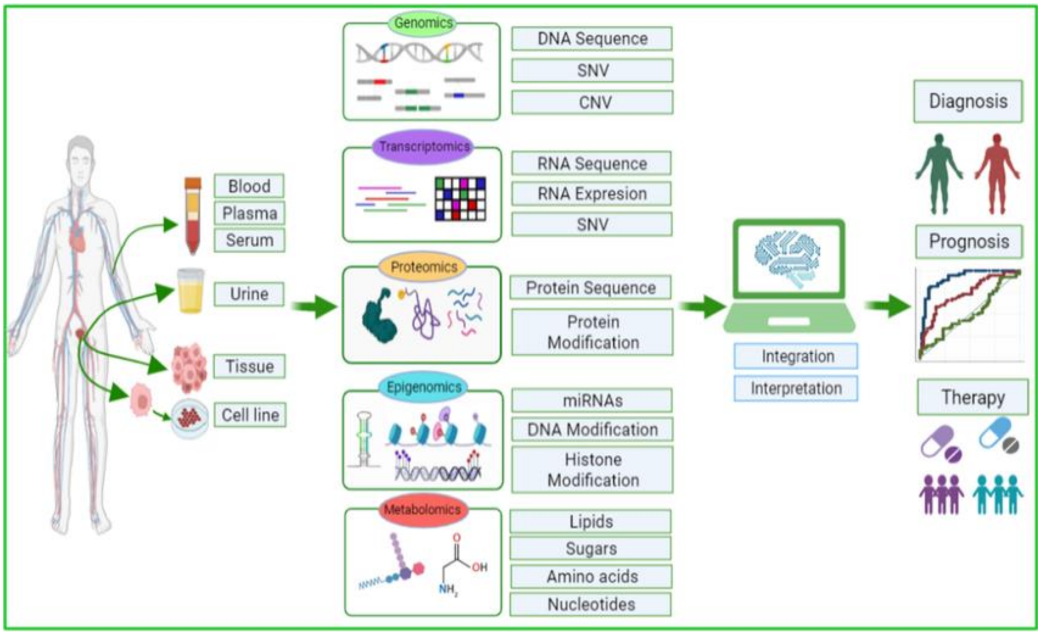
What is multi-omics data?

Multi-omics data refers to the integration and analysis of multiple types of “omics” data generated from biological systems, aiming to provide a comprehensive understanding of the underlying molecular mechanisms [7].

Each type of omics data captures information at different biological levels, enabling researchers to examine the relationships and interactions between these levels in a holistic manner.

Omics data types:

- *Gen**omics**:* focuses on DNA sequences, including genetic variations and mutations, to understand the genetic basis of diseases.
- *Transcript**omics**:* analyzes RNA expression levels to study gene activity and its regulation under various conditions.
- *Prote**omics**:* investigates protein expression, modifications, and interactions to explore cellular functions and signaling pathways.
- *Epigen**omics**:* studies DNA methylation, histone modifications, and chromatin structure to understand gene regulation mechanisms.
- *Metabol**omics**:* analyzes small molecules and metabolites to assess cellular metabolic states.
- *Microbi**omics**: ex*amines the composition and functions of microbial communities in relation to host health and disease.



Multi-omic strategy in Prostate cancer study

(source https://www.researchgate.net/publication/363281977_Prostate_cancer_in_omics_era)

Methods

Dataset Description

The dataset used in this study is the Prostate Adenocarcinoma (PRAD) dataset, a part of The Cancer Genome Atlas (TCGA) project [8].

TCGA is a large-scale genomics initiative that has collected and analyzed molecular data from over 11,000 cases across 33 tumor types. The project integrates various biological data types, including mRNA expression, miRNA expression, DNA copy number variations, DNA methylation, and protein expression, enabling comprehensive multi-omics analyses.

For this study, the PRAD dataset provides multi-omics profiles of prostate cancer patients. The analysis focuses on three omics data types: mRNA, miRNA, and protein expression data.

These data types were retrieved as a MultiAssayExperiment object [9] consisting of the following experiments:

(1) PRAD_miRNASeqGene-20160128 with 1,046 features (rows) and 547 samples (columns)
(2) PRAD_RNASeq2Gene-20160128 with 20,501 features and 550 samples
(3) PRAD_RPPAArray-20160128 with 195 features and 352 samples

What is a MultiAssayExperiment object?

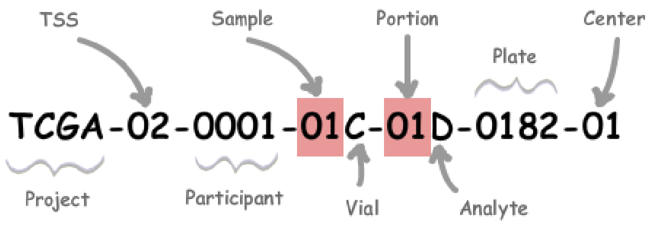
A MultiAssayExperiment object is a data structure designed to manage, integrate, and analyze multi-omics datasets collected from the same set of biological samples [9], such as patients in a clinical study.

The three key components of this data structure are:

- *ExperimentList:* a collection of individual assays, where each assay corresponds to a specific omics dataset (e.g., RNA-Seq, miRNA expression, methylation, or proteomics). Each assay is typically represented as a structured object, such as a SummarizedExperiment, ExpressionSet, or a matrix-like object - access **experiments()**
- *ColData:* it contains metadata about the biological samples (columns in the assays). It includes clinical and phenotypic information (e.g., patient ID, age, sex, tumor stage, treatment details) - access **colData()**
- *SampleMap:* a mapping table that links the sample identifiers in the assays to the unified biological sample identifiers (e.g., patient IDs) - access **sampleMap()**

Data Preprocessing

Each sample/patient is identified by a barcode with a specific structure:



Structure of TCGA barcodes (Image taken from here)

Components of the barcode:

- **Project Code:** indicates the data source or the project under which the sample was collected.
- **Tissue Source Site (TSS):** a two-character code identifying the site where the sample was collected, such as a specific hospital, clinic, or research institution.
- **Participant ID:** a four-character code uniquely identifying the patient.
- **Sample Type:** a three-character code denoting the type of sample (here the list with all sample types code) and its preparation.
- **Portion Number:** a two-digit number specifying the portion of the sample taken for analysis.
- **Analyte Type:** a single letter representing the type of analyte extracted (i.e., D: DNA, R: RNA, P: protein).
- **Plate ID:** a unique identifier for the plate used during sample processing.
- **Center Code:** a two-digit code denoting the sequencing or data generation center.

The first three components (Project Code, TSS, Participant ID) uniquely identify a specific individual, while the remaining components provide detailed metadata about the sample itself.

We use the barcode to:

- Retain only Primary Solid Tumors (identified by the sample type code "01") to have a more homogeneous group of samples.
- Exclude technical replicates in the dataset (i.e., repeated measurements from the same sample), identified by analyzing the Portion Number and other metadata.

The barcode facilitates the alignment and integration of data across different assays (e.g., RNA-seq, proteomics) and ensures consistent metadata annotation.

Then, other additional pre-processing steps were performed:

- *Removal of FFPE Samples:* Formalin-Fixed, Paraffin-Embedded (FFPE) samples are removed from the dataset, as these preservation methods are known to degrade data quality. Excluding FFPE samples ensures that only high-quality data is used in the analysis.
- *Retention of Complete Multi-Omics Data Samples:* only samples with data available across all the considered omics modalities (mRNA, miRNA, and protein expression) are retained. This step ensures that clustering and data integration methods are applied consistently across the full dataset, avoiding issues caused by missing values in any omics layer.

  The filtered dataset is then separated into individual matrices, with one matrix corresponding to each omics data type, and stored in a list for further processing.
- *Matrix Transposition:* each matrix is transposed such that samples are represented as rows and features (e.g., genes, proteins) are represented as columns. This format is required by most clustering and integration algorithms, which assume that rows correspond to individual samples and columns to features.
- *Removal of features with missing values:* features containing missing values (i.e., NA) are removed from the dataset. This approach is chosen over imputation since the presence of missing values is limited to a small subset of features within the proteomics data.
- *Feature selection:* to reduce the dimensionality of the dataset while retaining its most significant features, the top 100 features with the highest variance are selected from each data source (mRNA, miRNA, and protein expression). This feature selection strategy is widely used due to its simplicity and computational efficiency.

  However, it has some limitations:
  1. Univariate Nature: it evaluates features independently, ignoring interactions between them.
  2. Inability to Remove Redundancy: highly correlated or redundant features are not filtered out.
  3. Arbitrary Thresholds: choosing a fixed number of top features (e.g., 100) is inherently subjective and may not suit all datasets.
- Normalization: feature values are standardized using the z-score method, where each feature's values are transformed to have a mean of 0 and a standard deviation of 1. This ensures comparability across features by removing scale differences.
- Barcode Cleaning: TCGA sample barcodes are trimmed to retain only the consistent components: “Project-TSS-Participant”. This ensures uniformity across all omics datasets, enabling accurate matching and alignment of samples.

Data integration

Integrating different omics data remains a significant challenge in the scientific community, with numerous methods proposed to address this issue [10]. These methods aim to effectively combine heterogeneous data types to capture the complex biological processes underlying diseases.

To fuse our Prostate adenocarcinoma dataset, we employed a state-of-the-art approach called **Similarity Network Fusion** (SNF) [2]. This approach constructs and iteratively fuses similarity networks for each omics dataset, effectively capturing both shared and complementary information across data types.

In addition to SNF, we used a simple averaging method as a baseline integration approach. This method combines the similarity matrices from individual omics datasets through straightforward element-wise averaging, providing a reference point for evaluating the effectiveness of advanced integration techniques.

1. *Similarity Network Fusion*

The similarity metric used in SNF can vary, but the *scaled exponential Euclidean distance* is commonly employed due to its ability to emphasize small differences and manage scale variability across datasets.

Steps in SNF with Scaled Exponential Euclidean Distance:

a. *Compute Pairwise Distances*: for each data type, calculate the Euclidean distance between all pairs of entities *i* and *j*.

$$d(i,j) = \sqrt{\sum_{k=1}^{s} \quad (x_{ik} - x_{jk})}$$

b. *Apply the Scaling and Exponential Transformation:* transform the distances into similarities using the scaled exponential Euclidean distance formula.

$$W^{(s)}(i,j) \; = \; \exp\left(-\frac{d^2(i,j)}{\sigma^2}\right)$$

where:

- $d(i,j)$ is the Euclidean distance between two entities i and j .
- σ is a scaling parameter that controls the sensitivity of the similarity measure of the distances.

This transformation ensures that small distances lead to high similarity scores, while large distances are exponentially down-weighted, making it robust to outliers and noise.

c. *Construct Initial Similarity Networks:* each similarity matrix represents the relationships between entities based on one data type.

d. Derive Global and Local Similarity Matrices: construct the “global” similarity matrix $P(s)$ capturing the overall structure of relationships among samples.

   Construct the “local” similarity matrix $S(s)$, which emphasizes neighborhood-based relationships by focusing on the k-nearest neighbors N_i of each sample x_i.

e. *Iterative Fusion:* combine similarity networks iteratively, sharing and updating information between networks to create a unified, consensus similarity matrix.
   For two data modalities (s=2):

$$P_{t+1}^{(2)} = S^{(1)} \times P_t^{(2)} \times S^{(1)T}$$
$$P_{t+1}^{(1)} = S^{(2)} \times P_t^{(1)} \times S^{(2)T}$$

   Here, t is the iteration index.

f. *Converged Output:* the final fused network represents an integrated view of the relationships across all data types.

$$P^{(c)} = \frac{1}{s}\sum_{k=1}^{s} P^{(k)}$$

2. *Simple Average*

The similarity matrices computed from each omics layer were averaged to produce a single integrated similarity matrix.

Disease subtype discovery by clustering approaches

The integrative analysis of prostate adenocarcinoma (PRAD) has revealed seven distinct genomic alterations, which include fusions involving **ERG**, **ETV1**, **ETV4**, and **FLI1**, as well as mutations in **SPOP**, **FOXA1**, and **IDH1**.

Through the integration of genomic, transcriptomic, and epigenomic data, PRAD can be classified into three distinct molecular subtypes. This classification was achieved using the iCluster integrative clustering method [5], which combines multiple layers of omics data (somatic copy-number alterations, DNA methylation, mRNA, microRNA and Protein expression) to identify robust clusters that reflect key biological features of the disease.

The subtypes were defined through comprehensive molecular profiling of 333 primary prostate cancer samples from The Cancer Genome Atlas Research Network [6].

In this study, two clustering techniques were employed to uncover molecular subtypes: Partitioning Around Medoids (PAM) and Spectral Clustering.

Clustering algorithms description

1. *iCluster*

iCluster [5] is a statistical and computational method used for integrative clustering of multi-omics data.
It aims to identify clusters or subgroups within a dataset by simultaneously modeling multiple data types in a unified framework.

iCluster is based on a Gaussian latent variable model.
It assumes that the observed data $X^{(k)}$ (for k-th data type) is generated from a latent variable Z , which is shared across all data types.
The model can be summarized as:

$$X^{(k)} = ZW^{(k)} + \varepsilon^{(k)}$$

Where:

- $X^{(k)}$ the observed data matrix for the k-th data type.
- Z is the latent variable matrix representing the shared underlying structure.
- $W^{(k)}$ is the weight matrix that maps the latent variables to the observed data.
- $\varepsilon^{(k)}$ error term capturing noise.

2. *PAM algorithm*

The **Partitioning Around Medoids** (PAM) algorithm [3] [11] is a robust clustering technique that groups data into clusters using representative data points called *medoids*. Unlike centroids used in **k-means**, medoids are actual data points, making PAM particularly effective for non-Euclidean distance measures and datasets with noise or outliers.

PAM Pseudo-code:

```
Input: Data $X$ , number of clusters $k$ , distance metric.
Output: Final medoids and cluster assignments.
1. Initialize $k$ medoids $M = \{m_1, m_2, \ldots, m_k\}$ .
2. Repeat until convergence:
3. Assign each point $x_i$ to the closest medoid.
4. For each medoid $m$ , evaluate the cost of swapping $m$ with every non-medoid point.
5. Update $M$ if a swap reduces the total cost.
```

The algorithm operates in two main phases:

a. *BUILD PHASE:* initializes the clustering process by selecting k initial medoids.
   The computational complexity is $O(k \times n^2)$, as it involves evaluating the dissimilarities between all pairs of points during the medoid initialization.
b. *SWAP PHASE:* iteratively improves the medoids to further reduce the total clustering cost.
   The computational complexity is $O(k \times (n - k)^2)$, as it examines potential swaps between medoid and non-medoid points to identify configurations that reduce the overall clustering cost.

The primary goal of PAM is to minimize the average dissimilarity of the objects to their closest selected medoid. By doing so, it ensures that the chosen medoids represent the data clusters effectively.
PAM is computationally more expensive than k-means, primarily due to the $O\ (k \times (n - k)^2)$ complexity of the Swap Phase. This limits its scalability for large datasets.

The PAM algorithm was applied to:

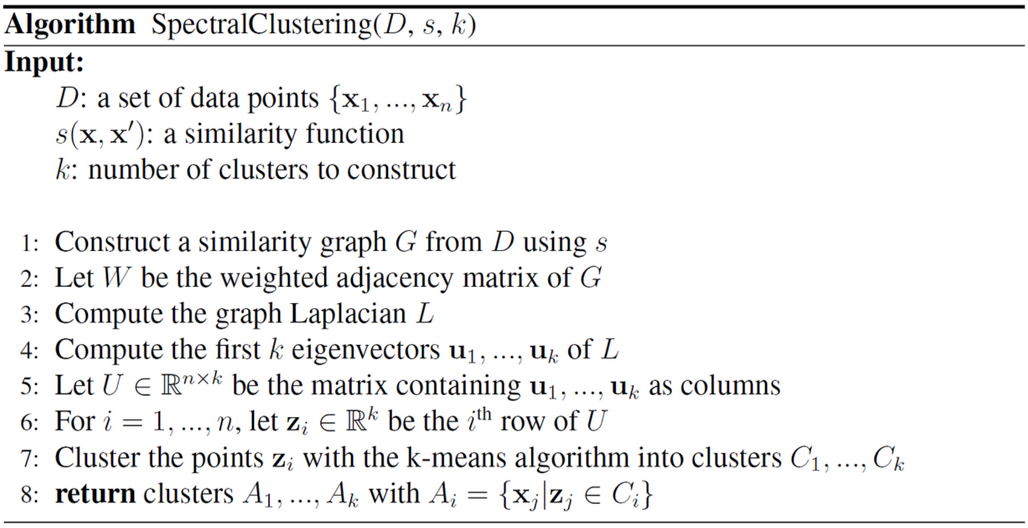
- Individual similarity matrices (mRNA, miRNA, and protein).
- The integrated matrix from the simple average method.
- The integrated matrix from SNF.

The number of clusters (k=3) was set to match the three iCluster subtypes identified by *The Cancer Genome Atlas Reasearch Network* [1].

3. *Spectral Clustering*

Spectral Clustering [4] [12] is graph-based clustering algorithm that uses the *eigenvalues* (spectrum) of a similarity matrix derived from the data to perform dimensionality reduction before clustering in fewer dimensions [13].
It is particularly effective for identifying clusters in non-convex and complex data distributions.

The main steps of the algorithm are showed below:



```
Algorithm SpectralClustering($D$, $s$, $k$)
Input:
    $D$: a set of data points $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$
    $s(\mathbf{x}, \mathbf{x}')$: a similarity function
    $k$: number of clusters to construct

Construct a similarity graph $G$ from $D$ using $s$
Let $W$ be the weighted adjacency matrix of $G$
Compute the graph Laplacian $L$
Compute the first $k$ eigenvectors $\mathbf{u}_1, ..., \mathbf{u}_k$ of $L$
Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing $\mathbf{u}_1, ..., \mathbf{u}_k$ as columns
For $i = 1, ..., n$, let $\mathbf{z}_i \in \mathbb{R}^k$ be the $i^{\text{th}}$ row of $U$
Cluster the points $\mathbf{z}_i$ with the k-means algorithm into clusters $C_1, ..., C_k$
return clusters $A_1, ..., A_k$ with $A_i = \{\mathbf{x}_j | \mathbf{z}_j \in C_i\}$
```

Spectral Clustering Algorithm (Image taken from here)

Advantages:

- Handles complex cluster shapes.
- Works with diverse similarity measures.

Disadvantages:

- Computing eigenvalues for large datasets can be expensive ($O(n^3)$).
- Sensitive to hyperparameters like similarity function and number of clusters.

The Spectral Clustering was applied on the integrated matrix obtained from Similarity Network Fusion.

The clustering results were evaluated by comparing them to the iCluster disease subtypes using three widely recognized and commonly used metrics for cluster comparison [14]:

- **Rand Index (RI):** measures the similarity between two clusterings X and Y by considering all pairs of samples and counting how often they are grouped in the same or different clusters in both clusterings.
  It is defined as:
  $$R(X,Y) = \frac{n_{11} + n_{00}}{\binom{n}{\cdot 2}}$$
  Where:
  - n_{11} is the number of pairs of samples placed in the same cluster in both clusterings.
  - n_{00} is the number of pairs of samples placed in different clusters in both clusterings.
  - n is the total number of samples.

  The RI ranges from 0 to 1, where higher values indicate greater agreement between the clusterings. A value of 1 signifies perfect concordance.

- **Adjusted Rand Index (ARI):** is an adjusted version of the Rand Index that accounts for the possibility of random chance clustering.
  While RI can be biased toward high values due to random clustering, ARI provides a more reliable comparison by normalizing the index.
  It is defined as:
  $$R_{adj}(X,Y) = \frac{R(X,Y) - E\,[R(X,Y)]}{\max\{R(X,Y)\} - E\,[R(X,Y)]}$$
  Where:
  - $E\,[R(X,Y)]$ is the expected RI value for random clusterings.
  - $\max\{R(X,Y)\}$ is the maximum possible value of the RI.

  The ARI ranges from –1 to 1, with 1 indicating perfect agreement, 0 indicating random clustering, and negative values indicating worse-than-random clustering.

- **Normalized Mutual Information (NMI):** measures the amount of information shared between two clusterings. It is a metric from information theory that quantifies how much knowing the clustering labels of one set of clusters reduces uncertainty about the other.

  NMI is calculated using the mutual information and the entropies of the two clusterings X and Y :

  $$NMI(X,Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}$$

  Where:

  - $I(X;Y)$ is the mutual information between the clusterings.
  - $H(X)$ and $H(Y)$ are the entropies of clusterings X and Y , respectively.

  The Mutual information is defined as:

  $$I(X;Y) = \sum_{x=1}^{k}\sum_{y=1}^{l} p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)}$$

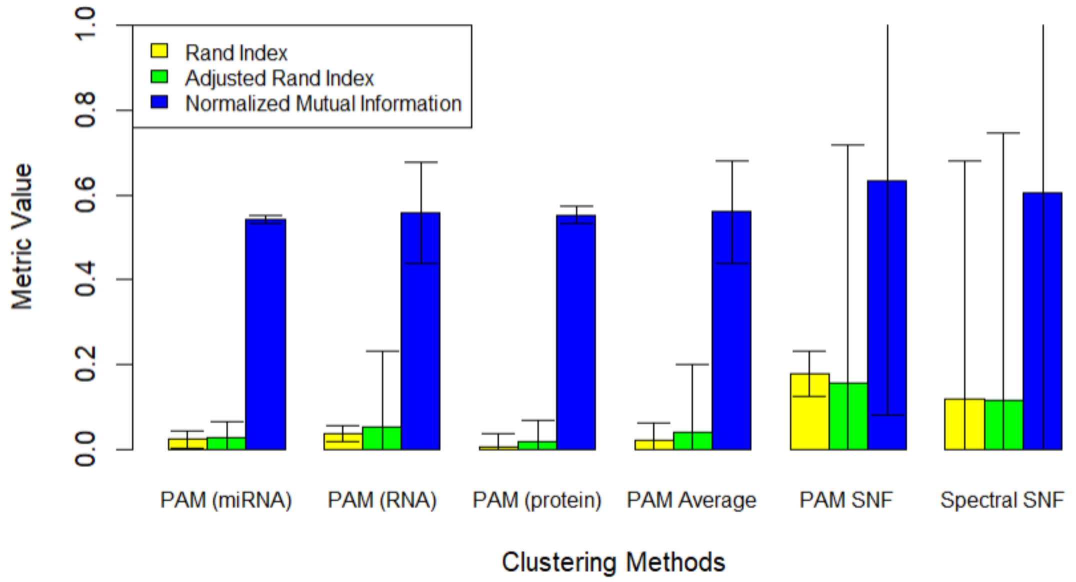
  Where:

  - $p(x,y)$ is the joint probability of x and y .
  - $p(x)$ and $p(y)$ are the marginal probabilities of x and y , respectively.

  The NMI ranges from 0 to 1, with higher values indicating a greater degree of similarity between the clusterings. A value of 1 represents perfect agreement.

Results

	Rand Index	Adjusted Rand Index	Normalized Mutual Information
PAM Single	0.024669347	0.02763638	0.5424122
PAM Single	0.037660032	0.05320068	0.5574964
PAM Single	0.007886092	0.01965598	0.5523704
PAM Average	0.023650868	0.04030343	0.5598145
PAM SNF	0.179466272	0.15674451	0.6316769
Spectral SNF	0.119098119	0.11723454	0.6051326



PAM SNF appears to perform the best across all three metrics, especially in NMI (0.6317). This suggests that integrating features from multiple data types (such as miRNA, RNA, and protein) using SNF yields the most consistent and informative clusters in comparison to the true disease subtypes.

Spectral SNF also performs well in terms of NMI (0.6051), but it lags behind PAM SNF and others in the Adjusted Rand Index and Rand Index. This might suggest that the spectral clustering approach could capture some structure, but with slightly less precision.

PAM Single (RNA) stands out for its relatively stronger performance in the Adjusted Rand Index (0.0532), though the overall metrics suggest that it's still far from perfect.

PAM Single (miRNA) and **PAM Single (protein**) have the lowest performance in all three metrics, especially for the Rand Index and Adjusted Rand Index, suggesting that these data sources alone may not be sufficient to accurately capture the disease subtypes.

References

[1] G. Tini, L. Marchetti, C. Priami, and M. P. Scott-Boyer (2022). “Multi-omics integration—a comparison of unsupervised clustering methodologies”. *Briefings in Bioinformatics*, vol 23, no. 1, p. 547.

[2] B. Wang et al. (2014). “Similarity network fusion for aggregating data types on a genomic scale”. Nature methods, vol. 11, no. 3, pp. 333–337, 2014.

[3] John Wiley & Sons (1990). “Partitioning around medoids (program PAM)”. *Finding groups in data*, pp. 68-125. https://onlinelibrary.wiley.com/doi/10.1002/9780470316801.ch2.

[4] U. Von Luxburg (2007). “A tutorial on spectral clustering”. Statistics and computing, vol. 17, pp. 395–416.

[5] R. Shen, A.B. Olshen, and M. Ladanyi (2009). “Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis”. Bioinformatics, vol. 25, no. 22, pp. 2906-2912.

[6] A. Abeshouse et al. (2015). "The molecular Taxonomy of primary prostate cancer". Cell, vol 163, no. 4, pp. 1011-1025.

[7] Y. Hasin, M. Seldin, and A. Lusis (2017). “Multi-omics approaches to disease”. Genome Biology, vol. 18, no. 1, p. 83.

[8] C. Hutter and J. C. Zenklusen (2018). “The cancer genome atlas: Creating lasting value beyond its data”. Cell, vol. 173, no. 2, pp. 283–285.

[9] M. Ramos et al. (2017). “Software for the integration of multiomics experiments in bioconductor”. Cancer research, vol. 77, no. 21, pp. e39–e42.

[10] J. Gliozzo et al. (2022). “Heterogeneous data integration methods for patient similarity networks”. Briefings in Bioinformatics, vol. 23, no. 4, p. bbac207.

[11] H. Mushtaq and S.G. Khawaja (2018). “A Parallel Architecture for the Partitioning Around Medoids (PAM) Algorithm for Scalable Multi-Core Processor Implementation with Applications in Healthcare”. Sensors, vol. 18, no.12, p. 4129.

[12] A.Y. Ng, M.I. Jordan, Y. Weiss (2002). “On spectral clustering: Analysis and an algorithm”. Advances in neural information processing systems, pp. 849– 856.

[13] Wikipedia, “Spectral clustering — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/wiki/Spectral_clustering, 2025.

[14] S. Wagner and D. Wagner (2007). Comparing clusterings: An overview. Universität Karlsruhe, Fakultät für Informatik Karlsruhe.