Disease Subtype Discovery using Multi-Omics Data Integration

Ali Tabaraei
Computer Science Department
University of Milan
Milan, Italy
ali.tabaraei@studenti.unimi.it

Jessica Gliozzo
Computer Science Department
University of Milan
Milan, Italy
jessica.gliozzo@unimi.it

Giorgio Valentini
Computer Science Department
University of Milan
Milan, Italy
valentini@di.unimi.it

***Abstract*—Disease subtype discovery plays a crucial role in personalized medicine, aiming to identify homogeneous patient cohorts with similar clinical and molecular profiles, which can lead to enhancement in prognostic predictions and optimized treatment strategies for each patient's unique profile. The objective of this study is to investigate the effectiveness of different clustering approaches for disease subtype discovery, leveraging multi-omics data integration techniques. The dataset used comprised molecular data from diverse omics sources, including mRNA, miRNA, and protein expression data. We present the clustering results and compare them with known disease subtypes in the literature to evaluate the performance of the methods. In short, our findings indicate that leveraging PAM and Spectral clustering methods, coupled with the integration of multi-omics data using the integrated matrix by SNF, yielded the highest performance compared to other techniques investigated.**

***Index Terms*—Multi-Omics, Disease Subtype, Integration**

I. Introduction

In recent years, advancements in measurement technologies have enabled the collection of vast amounts of multi-omics data, ranging from genomic and epigenomic profiles to transcriptomic and proteomic measurements. These datasets contain valuable information about the molecular signatures underlying complex diseases, offering opportunities for understanding disease mechanisms, identifying potential therapeutic targets, and developing personalized treatment strategies [1].

Due to the paradigm shift towards personalized medicine [2], there has been growing interest in leveraging multi-omics data integration techniques and advanced clustering algorithms to unravel the heterogeneity of diseases and uncover hidden subtypes within patient populations [3]. One of the key challenges in this endeavor is the integration and analysis of diverse data modalities to uncover meaningful patterns and associations [4], [5].

Traditional approaches often involve the independent analysis of each data type, or using manual integration, which may overlook important dependencies present in the data [3], [5]. To address this challenge, novel computational methods have emerged, aiming to integrate multi-omics data and extract insights for precision medicine applications [6]. These methods leverage advanced statistical and machine learning techniques to model the complex relationships between different omics and uncover hidden structures within the data [3].

In this study, we aim to address these challenges by employing state-of-the-art data integration techniques and clustering algorithms to explore disease subtype discovery using multi-omics data. Section II will describe our methodology for data preprocessing, integration, and clustering, as well as the metrics used to evaluate the performance of our approach. Furthermore, in the section III we will present the results of our analysis and discuss the implications of our findings. Lastly, in section IV, we discuss the potential shortages of our analysis, as well as highlight avenues for future research in the field of multi-omics data analysis and precision medicine.

II. Methods

A. Dataset

The project focuses on uncovering disease subtypes using a multi-omics dataset sourced from The Cancer Genome Atlas (TCGA) program [7]. TCGA represents a genomics initiative that contains over 11,000 cases spanning 33 tumor types, incorporating diverse biological data sources such as mRNA expression, miRNA expression, copy number values, DNA methylation, and protein expression.

Specifically, we utilize the `curatedTCGAData` [8] package for our analysis to work with the *Prostate adenocarcinoma dataset* (disease code "PRAD"), considering only 3 different omics data sources (miRNA, mRNA, and protein expression data), as they were investigated by *The Cancer Genome Atlas Research Network* [9] and their integrative clustering model (called *iCluster* [4]).

Prostate cancer is one of the most prevalent cancers among men worldwide, with considerable variability in its molecular characteristics and clinical behavior. Despite advancements in risk stratification using clinical and pathological parameters, current tools often fall short in accurately predicting disease outcomes. Molecular profiling has emerged as a promising approach to further identify prostate cancers based on their underlying genetic alterations, potentially distinguishing disease subtypes [9].

B. Preprocessing

Before delving into the clustering algorithms, it is imperative to preprocess and cleanse the data to ensure optimal performance and reliability throughout the analysis. The following steps are taken sequentially:

1) **Select primary tumor types:** We aim to include only samples of patients with a *primary* tumor type (excluding metastases, which constitute abnormal masses) to ensure having a homogeneous group of samples. According to the TCGA barcode structure, we will select samples with "Primary Solid Tumors" which are identified by the code `01` in the *sample* part of the barcode.
2) **Exclude duplicated samples:** We need to examine whether technical replicates are present and exclude any duplicated patient samples if identified. To accomplish this, we can leverage the first 12 characters of the barcode to uniquely identify patients and subsequently filter out any duplicated samples associated with them. In this specific dataset, no replicated data exists.

3) **Remove samples preserved using FFPE:** There are two primary tissue preparation methods to store and preserve samples: *FFPE* (Formalin-Fixed Paraffin-Embedded), and *freezing* the samples. Due to the superior preservation of DNA and RNA molecules in frozen tissues, samples preserved using the FFPE technique will be excluded from further analysis, which in our case there were none to be removed.
4) **Select samples having all omics sources:** Not every sample has all the omics data sources available. Therefore, we will limit our analysis to samples that possess data for all the considered omics, resulting in the same number of samples for all omics sources.
5) **Ensure having features in columns:** In the majority of Bioinformatics data sources, features are typically arranged in rows, with samples represented as columns in the matrix. To align with conventional data science practices, we will transpose the matrices, ensuring to have features in columns and samples in rows.
6) **Remove features with missing values:** Before performing any type of machine learning algorithm, the data should be free of possible existing missing values. In this study, since only a negligible number of features in proteomics data contain missing values, we will directly remove those features rather than using permutation techniques.
7) **Select the top 100 features having highest variance:** In the field of Bioinformatics, the existence of a higher number of features than samples can pose challenges for machine learning techniques, and it may lead to poor performance particularly when numerous features with significant contributions have low variance. In this study, as asked by the project description, we will only select the top 100 features having the highest variance from each data source. However, this strategy overlooks feature interactions and redundancy, introducing potential limitations.
8) **Normalize the data:** Since the omics data have been acquired with diverse measurements in various units or scales, we can perform standardization techniques to facilitate a meaningful comparison and analysis. To achieve this, we will use *z-score normalization*, which guarantees a standard normal distribution with a mean of 0 and a standard deviation of 1. The formula (1) is provided below, where μ represents the mean and σ corresponds to the standard deviation.

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

9) **Clean the barcode names:** The final step in our preprocessing pipeline is to clean the barcodes to retain only the first 12 characters for each individual, i.e. "Project-TSS-Participant".

C. Disease subtypes

To compare our clustering results with the literature, we utilize the disease subtypes reported by PAM50 [10].

1) First, we download the disease subtypes having the `PRAD` code as cancer type (Prostate adenocarcinoma), using the `TCGAbiolinks` package, where the disease subtype clusters of iCluster are stored in the `Subtype_Integrative` column.
2) Then, we filter out the samples of patients without a primary tumor type. Similar to the final preprocessing step, we retain only the first 12 characters of the barcode for each individual existing in the multi-omics dataset.

TABLE I
DIFFERENT SUBTYPES AND THEIR COUNT INVESTIGATED BY ICLUSTER

iCluster.Subtype	Count
1	60
2	83
3	105

3) Next, we check that the patients in the multi-omics dataset and subtypes are in the same order. This is a vital step in our implementation to avoid further errors.
4) Finally, keep in mind that in our selected subset of samples (having all three data sources available), not every sample has an associated subtype. Therefore, we eliminate such samples by comparing the barcodes of subtypes and multi-omics dataset.

Table I demonstrates the resulting number of samples for each subtype.

D. Data Integration

Several different strategies have been studied by the literature to mix and combine the different multi-omics data sources. In this paper, we implement three of these strategies, which are described as follows.

- **Integrating the data using SNF:** Recent advancements in data collection allow cost-effective gathering of diverse genome-wide data. *Similarity Network Fusion* (SNF) [5], efficiently combines these data types, creating a comprehensive view of diseases or biological processes. SNF constructs individual networks for each data type (e.g., mRNA expression, DNA methylation) and merges them into a unified network. This approach is exemplified in the analysis of patient cohorts, where SNF computes and fuses patient similarity networks from each data type, providing a concise yet comprehensive understanding of underlying biological mechanisms. The algorithm is described in the following:
  1) First, the similarity matrix among samples of each data source s (miRNA, mRNA, and protein expression data) is computed separately based on their gene expression profiles. We use the *scaled exponential Euclidean distance* [5] as the similarity measure:

$$W(i,j) = exp\left(-\frac{\rho(x_i, x_j)^2}{\mu\varepsilon_{ij}}\right) \tag{2}$$

  where $\rho(x_i, x_j)$ is the Euclidean distance between patients x_i and x_j, μ is a parameter, and $\varepsilon_{i,j}$ is a *scaling factor* defined as (3), having $mean(\rho(x_i, N_i))$ as the average value of the distances between x_i and each of its neighbors.

$$\begin{aligned} \rho_1 &= \text{mean}(\rho(x_i, N_i)) \\ \rho_2 &= \text{mean}(\rho(x_j, N_j)) \\ \rho_3 &= \rho(x_i, x_j) \\ \varepsilon_{i,j} &= \frac{\rho_1 + \rho_2 + \rho_3}{3} \end{aligned} \tag{3}$$

  2) A *global* similarity matrix $P^{(s)}$ is derived from $W^{(s)}(i,j)$, capturing the overall relationships between samples:

$$P^{(s)}(i,j) = \begin{cases} \frac{W^{(s)}(i,j)}{2\sum_{k\neq i} W^{(s)}(i,k)} & \text{, if } j \neq i \\ 1/2 & \text{, if } j = i \end{cases} \quad (4)$$

3) A *local* similarity matrix $S^{(s)}$ is derived from $W^{(s)}(i,j)$, capturing the local structure of the network based on local similarities in the neighborhood (defined as $N_i = \{x_k | x_k \in kNN(x_i) \cup \{x_i\}\}$) of each individual, and setting to zero all the others:

$$S^{(s)}(i,j) = \begin{cases} \frac{W^{(s)}(i,j)}{\sum_{k\in N_i} W^{(s)}(i,k)} & \text{, if } j \in N_i \\ 0 & \text{, otherwise} \end{cases} \quad (5)$$

4) Through an iterative process, given s data sources (here $s = 3$), s different W, S and P matrices are constructed where similarities are diffused through the Ps until convergence (matrices P become similar). To achieve this, for each different s, P is updated by using S from the same data source but P from a different view, and vice versa. In the simplest case, when $s = 2$, we have $P_t^{(s)}$ that refers to P matrices for data $s \in \{1, 2\}$ at time t. In this case, the following recursive updating formulas describe the diffusion process:

$$\begin{aligned} P_{t+1}^{(1)} &= S^{(1)} \times P_t^{(2)} \times S^{(1)\top} \\ P_{t+1}^{(2)} &= S^{(2)} \times P_t^{(1)} \times S^{(2)\top} \end{aligned} \quad (6)$$

5) The final *integrated* matrix $P^{(c)}$ is computed by averaging as below:

$$P^{(c)} = \frac{1}{s} \sum_{k=1}^{s} P^{(k)} \quad (7)$$

For the implementation of the multi-omics data integration using SNF (derived from CRAN `SNFtool` package), we set the number of iterations $t = 20$, and the number of neighbors $K = 20$ to be considered for the local similarity matrix $S^{(s)}$ computation.

- **Integrating the data using simple averaging:** This is the most trivial multi-omics data integration strategy that can be utilized to fuse the different similarity matrices from each data source into one, by performing a simple averaging of the matrices as (8):

$$W_{\text{avg}}(i,j) = \frac{1}{|\text{data sources}|} \sum_{s\in\text{data sources}} W^{(s)}(i,j) \quad (8)$$

- **Integrating the data using NEMO:**
NEMO [11], standing for *NEighborhood based Multi-Omics clustering*, is a novel algorithm for multi-omics clustering. NEMO can be applied to partial datasets in which some patients have data for only a subset of the omics, without performing data imputation. It should be noted that the `nemo.affinity.graph()` function for data integration takes as input a matrix with features in rows and samples in columns, so we need to transpose the input to this function.

E. Clustering (disease subtype discovery)

We will now proceed with the implementation of a clustering algorithm to identify the disease subtypes, which we can later compare these obtained clusters with the disease subtypes investigated by iCluster.

- **PAM clustering:** PAM [10] clustering algorithm, short for *partition around medoids*, aims to identify a set of candidate medoids that represent the center of the clusters, minimizing the average dissimilarity of objects to their closest selected medoid by iteratively selecting and swapping medoids. The process continues until no further improvement can be made, and is divided into two main phases:
  1) **Build Phase:** We explicitly set k as the number of clusters to be found, and we attempt to find the candidate medoids to be stored in set S. We initialize S by adding an object with minimal distances to all other objects. Then, to add other $k-1$ elements to S, we perform the following steps iteratively to find each candidate medoid:
     a) Let's set O as the set of *all* objects, S as the *selected* objects, and $U = O - S$ as *unselected* objects. Considering a new candidate $i \in U$, for all the other unselected objects $j \in U - \{i\}$, first compute the distance between j and the closest medoid currently in S, namely D_j, and then compute the distance between j and the new candidate i, namely $d(i,j)$.
     b) The clustering may benefit from the new candidate i if $d(i,j) < Dj$, so will aggregate the contribution of all j into a total gain computed as $g_i = \sum_{j\in U} max\{(D_j - d(i,j)), 0\}$. Consequently, we will choose the candidate i maximizing g_i, and update the $S := S \cup \{i\}$ and $U := U - \{i\}$ accordingly.
     c) We repeat until k candidates have been selected.
  2) **Swap Phase:** It attempts to improve the quality of the selected candidates by swapping objects between S and U. For each pair $(i,h) \in S \times U$ (where $i \in S$ and $h \in U$) to be considered for swapping, we perform the following steps:
     a) We swap i and h, as h becomes a candidate and i is unselected.
     b) For each object $j \in U - \{h\}$ (all except the swapped objects), if $d(j,i) > D_j$ (where D_j is the dissimilarity between j and the *closest* object in S), then we compute the contribution $K_{jih} = min\{d(j,h) - D_j, 0\}$. Otherwise, if $d(j,i) = D_j$, then, $K_{jih} = min\{d(j,h), E_j\} - D_j$ (where E_j is the dissimilarity between j and the *second closest* object in S) is computed.
     c) We compute the total result of the swap as $T_{ih} = \sum\{K_{jih} | j \in U\}$, and we select the pair (i,h) minimizing T_{ih}.
     d) If $T_{ih} > 0$, the algorithm halts since the objective value cannot be decreased. Otherwise, if $T_{ih} < 0$, the swap is performed, D_j and E_j are updated, and we jump to the first step of the "Swap" phase.

  In this study, we set the number of clusters to be found by the PAM algorithm as $k = 3$, which is equal to the number of disease subtypes found by iCluster. Then, we perform this algorithm on the following similarity matrices:
  1) Similarity matrices obtained from each single data source. The resulting clusterings are named `PAM_miRNA`, `PAM_mRNA`, and `PAM_protein`, respectively.
  2) Integrated matrix obtained from averaging over matrices, where the resulting clustering is named `PAM_W_avg`

3) Integrated matrix obtained from SNF, where the resulting clustering is named `PAM_SNF`
4) Integrated matrix obtained from NEMO, where the resulting clustering is named `PAM_AF_NEMO`

- **NEMO clustering:** NEMO provides the possibility of performing clustering using another approach called *Spectral Clustering* [12]. We use the function `nemo.clustering()` to test this approach. We name the resulting clustering as `NEMO`.
- **Spectral clustering:** We perform *Spectral Clustering* on the integrated matrix obtained from Similarity Network Fusion (SNF) utilizing `spectralClustering()`, where the features are placed in rows and samples in columns, so we need to transpose the input to this function. We name the resulting clustering as `Spectral`.

F. Evaluation Metrics

In the literature, several evaluation metrics can be found to compare the clustering results [13], and using `mclustcomp` R package we can access 24 different scores. Here, we will consider 3 of the most used techniques, which are described in the following:

- **Rand Index (RI):** Given the clusters C_1 and C_2, this metric can be computed by counting the pair of objects in the same cluster in both C_1 and C_2 (denoted as n_{11}), along with the pair of objects in different clusters both in C_1 and C_2 (denoted as n_{11}), concerning the all possible pairs. As a result, this metric is bounded within $[0, 1]$, representing similar clusters when $R(C_1, C_2)$ approaches 1, and dissimilar clusters when near zero.

$$R(C_1, C_2) = \frac{2(n_{11} + n_{00})}{n(n-1)} \tag{9}$$

- **Adjusted Rand Index (ARI):** The *Adjusted Rand Index* (ARI) is utilized to measure the similarity between two data clusterings. It is an enhancement over the Rand Index as a basic measure of similarity between two clusterings, overcoming its disadvantage of being sensitive to chance. The ARI takes into account the fact that two random partitions of a dataset should not assume a constant value, and it adjusts the Rand Index to account for this possibility. ARI ranges within $[-0.5, 1]$, where it represents identical clustering on values near 1, and indicates independent clusterings when approaching negative values.
- **Normalized Mutual Information (NMI):** The *Mutual Information* (MI) measures how much we can reduce uncertainty about an element's cluster when we already know its cluster in another clustering. It is defined as (10), where $P(i, j) = \frac{|C_{1i} \cap C_{2j}|}{n}$ is the probability that an element belongs to cluster $C_i \in C_1$ and cluster $C_j \in C_2$:

$$MI(C_1, C_2) = \sum_{i=1}^{k} \sum_{j=1}^{l} P(i, j) log_2 \frac{P(i, j)}{P(i)P(j)} \tag{10}$$

However, since it is not upper-bounded, it would be difficult to interpret the obtained results, so we use a normalized version of MI to bound it in the range $[0, 1]$ (maximum NMI for identical clusterings), which is *Normalized Mutual Information* (NMI). It is defined as (11), where $H(C_1)$ and $H(C_2)$ are the corresponding entropies of the clusterings C_1 and C_2:

$$NMI(C_1, C_2) = \frac{MI(C_1, C_2)}{\sqrt{H(C_1)H(C_2)}} \tag{11}$$

TABLE II
CLUSTERING RESULTS FOR DIFFERENT METRICS

Clustering	Metric		
	RI	ARI	NMI
PAM_miRNA	0.5424122	0.024669347	0.02763638
PAM_mRNA	0.5574964	0.037660032	0.05320068
PAM_protein	0.5523704	0.007886092	0.01965598
PAM_W_avg	0.5598145	0.023650868	0.04030343
PAM_SNF	0.6316769	0.179466272	0.15674451
PAM_AF_NEMO	0.5617735	0.032536154	0.05438288
NEMO	0.3803056	0.014593726	0.06891715
Spectral	0.6051326	0.119098119	0.11723454

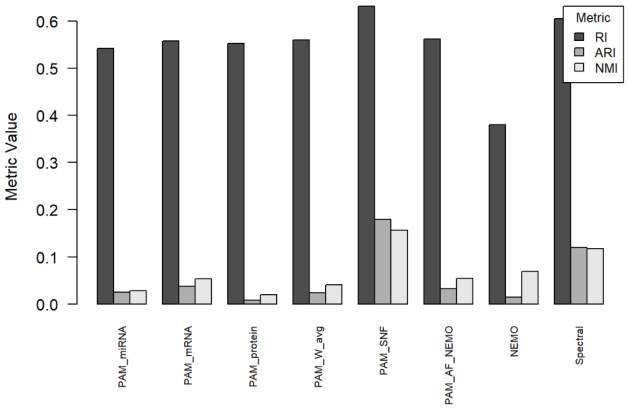

Fig. 1. Performance of Clustering Techniques (by clustering method)

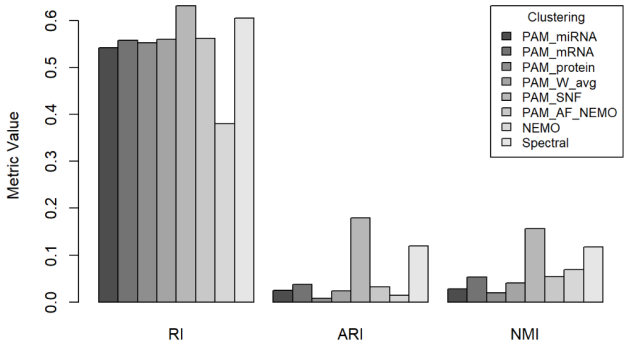

Fig. 2. Performance of Clustering Techniques (by evaluation metric)

III. RESULTS

Now that we have different clustering results obtained from PAM, NEMO, and Spectral clustering, along with different choices of matrix integration for PAM, we can proceed with comparing the aforementioned results with the disease subtypes reported by iCluster.

First, we will compute the evaluation metrics RI, ARI, and NMI for all the clustering algorithms proposed compared to the PAM50. Table II highlights the different values obtained for all different metrics.

With the evaluation metric values available, we proceed to visualize the results in Fig. 1 and Fig. 2.

Analyzing the results of clustering algorithms compared to PAM50, the following observations can be made:

- Overall, it can be seen that performing the PAM algorithm on the SNF integrated matrix has the best performance among

the techniques explored, followed by the Spectral clustering approach on the same integrated matrix, which also provides promising results.
- It is evident that the RI metric provides more optimistic results, so it would be a valid idea to present other measures such as ARI (based on counting pairs) and NMI (based on information theory) to better interpret the results.
- Due to the simplicity of the chosen preprocessing techniques (such as selecting the first 100 highest variance, or selecting only the samples having all the 3 data sources available instead of imputing), it is probable that some of the aforementioned group of genes are removed. This could potentially complicate the identification of subgroups based on alternative features and result in poor performance.
- PAM50 solely relies on mRNA data from different genes, whereas our clustering approach integrates multiple data sources from three distinct omics. Consequently, we have a richer dataset for clustering analysis, potentially resulting in clusters with unique biological interpretations that differ from the ones reported by PAM50.

IV. Discussion

The obtained results from our analysis unveil several insights into the challenges and opportunities in disease subtype discovery using multi-omics data integration. While our approach exhibited promising potential, especially with techniques like PAM and Spectral clustering coupled with multi-omics integration using SNF, the overall performance fell short of expectations. We highlight the following observations:

- **Limitations of preprocessing:** Firstly, the preprocessing steps applied to the multi-omics data may have been overly simplistic, leading to the loss of important information and introducing noise into the analysis. Naive data preprocessing methods may not adequately capture the complexities inherent in multi-omics datasets, thereby limiting the effectiveness of subsequent clustering algorithms.
- **Challenges of multi-omics data integration:** Moreover, while multi-omics integration holds promise for uncovering novel disease subtypes and identifying underlying biological mechanisms, it also introduces increased computational complexity and potential confounding factors. The heterogeneity and high-dimensional nature of multi-omics data present significant challenges in accurately capturing the underlying structure of the data and identifying meaningful clusters.
- **Comparison with PAM50:** Integrating diverse omics data sources introduces additional complexities in data normalization, feature selection, and algorithm parameter tuning, which may not have been fully addressed in our analysis. As a result, comparing such integrated matrix from multi-omics sources with the PAM50 results comprising only mRNA expression data poses its own set of challenges.
- **Insights for future research:** In conclusion, while our study provides valuable insights into the potential of multi-omics data integration for disease subtype discovery, the modest performance of our approach underscores the need for more sophisticated preprocessing techniques and algorithmic frameworks tailored to the complexities of multi-omics data. Future research efforts should focus on developing robust methods for data preprocessing, feature selection, and clustering analysis to unlock the full potential of multi-omics data in precision medicine applications.

References

[1] I. R. König, O. Fuchs, G. Hansen, E. von Mutius, and M. V. Kopp, "What is precision medicine?" *Eur Respir J*, vol. 50, no. 4, p. 1700391, 2017.

[2] S. J. Aronson and H. L. Rehm, "Building the foundation for genomics in precision medicine," *Nature*, vol. 526, no. 7573, pp. 336–342, 2015.

[3] J. Gliozzo, M. Mesiti, M. Notaro, A. Petrini, A. Patak, A. Puertas-Gallardo, A. Paccanaro, G. Valentini, and E. Casiraghi, "Heterogeneous data integration methods for patient similarity networks," *Briefings in Bioinformatics*, vol. 23, no. 4, p. bbac207, 2022.

[4] R. Shen, A. B. Olshen, and M. Ladanyi, "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, 2009.

[5] B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, B. Haibe-Kains, and A. Goldenberg, "Similarity network fusion for aggregating data types on a genomic scale," *Nature methods*, vol. 11, no. 3, pp. 333–337, 2014.

[6] G. Nicora, F. Vitali, A. Dagliati *et al.*, "Integrated multi-omics analyses in oncology: a review of machine learning methods and tools," *Frontiers in Oncology*, vol. 10, p. 1030, 2020.

[7] C. Hutter and J. C. Zenklusen, "The cancer genome atlas: creating lasting value beyond its data," *Cell*, vol. 173, no. 2, pp. 283–285, 2018.

[8] M. Ramos, L. Geistlinger, S. Oh, L. Schiffer, R. Azhar, H. Kodali, I. de Bruijn, J. Gao, V. J. Carey, M. Morgan *et al.*, "Multiomic integration of public oncology databases in bioconductor," *JCO Clinical Cancer Informatics*, vol. 1, pp. 958–971, 2020.

[9] A. Abeshouse, J. Ahn, R. Akbani, A. Ally, S. Amin, C. D. Andry, M. Annala, A. Aprikian, J. Armenia, A. Arora *et al.*, "The molecular taxonomy of primary prostate cancer," *Cell*, vol. 163, no. 4, pp. 1011–1025, 2015.

[10] *Partitioning Around Medoids (Program PAM)*. John Wiley and Sons, Ltd, 1990, ch. 2, pp. 68–125. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470316801.ch2

[11] N. Rappoport and R. Shamir, "Nemo: cancer subtyping by integration of partial multi-omic data," *Bioinformatics*, vol. 35, no. 18, pp. 3348–3356, 2019.

[12] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, pp. 395–416, 2007.

[13] S. Wagner and D. Wagner, *Comparing clusterings: an overview*. Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.