Disease subtype discovery using multi-omics data integration

Bioinformatics exam project

Alessia Cecere, Alessandro Di Gioacchino

June 2023

1 Introduction

The ongoing decrease in cost and processing time of omics-based methodologies has led to a significant increase in the volume of big data, shifting research methodologies from hypothesis-driven investigations to **data-driven analyses**. However, **single-level omics** approaches have limitations in establishing causal relationships between molecular alterations and phenotypic manifestations [16]. To achieve a comprehensive understanding of human health and diseases, it is necessary to interpret the intricate molecular complexities and variations across multiple levels, including the *genome*, *epigenome*, *transcriptome*, *proteome*, and *metabolome*. The recollection of data at these various levels is referred to as **multi-omics data** [15].

In recent times, various studies have shown that combining omics datasets yields a better understanding of the system under study: this is particularly true when unraveling aberrant cellular functions behind complex diseases, such as cancer [10]. Furthermore, a prominent challenge in the context of cancer research lies in the remarkable degree of progression **differences** in affected individuals, often depending on many factors such as environment and lifestyle.

One of the main applications of multi-omics data is the process of identifying the **underlying subtypes** of a **disease** by classifying samples into known subgroups, to understand disease etiology for different subtypes. This is an integrative part of what is called **personalized medicine**. Personalized medicine – also known as precision medicine – is an emerging approach for individualizing the practice of medicine, taking into account individual variability in predicting disease progression and transitions between disease stages, with the goal of selecting the most appropriate medical interventions [8].

In this context, the biomedical discovery of disease subtypes via unsupervised clustering of multiple sources is challenged by the inter-dependency of these sources, and their **heterogeneity**. Omics data use can be approached from two standpoints, involving a bottom-up and a top-down integration strategy [13]: in the first approach, multiple data types are combined first, followed

by manual integration of separate clusters; in the latter, data integration and dimensionality reduction are applied at the same time. According to [15] – although many tools use a combination of approaches – **data integration algorithms** can be broadly classified in the following **categories**.

- **Similarity-based methods**, like *Perturbation clustering for data integration and disease subtyping* (PINSPlus) [9] and *Neighborhood-based multi-omics clustering* (NEMO) [14]. These approaches focus on measuring the similarity and dissimilarity between samples based on their multi-omics profiles.
- **Fusion-based methods**, like *Pattern Fusion Analysis* (PFA) [12], aiming at integrating and fusing the data sources to create a unified representation of different omics layers.
- **Network-based methods**, capturing the inherent relationships and interactions between molecules, to uncover biological insights and patterns. An example of these approaches is *Similarity Network Fusion* (SNF) [6].
- **Bayesian methods**, like the *Pathway Recognition Algorithm using Data Integration on Genomic Models* (PARADIGM) [3], *iCluster* [2] and *iCluster+* [5].
- **Multivariate methods**, like *mixOmics* [11] and *Joint and individual variation explained* (JIVE) [4].

In this work, we applied a subset of these approaches to perform disease subtype discovery in the context of **prostate cancer**.

2 Methods

2.1 The Dataset

Prostate adenocarcinoma is a multi-omics dataset collected by the *The Cancer Genome Atlas* (TCGA) [18] program. **TCGA** houses one of the largest collections of multi-omics data sets, for more than 33 different types of cancer and 20000 individual tumor samples [4]; the aim of the initiative is to generate, merge, analyze, and interpret the DNA, RNA, protein and epigenetic profile changes in tumor samples, together with the clinical and histological data [15].

From a more technical standpoint, we used the `curatedTCGAData` package to download the data views we were interested in, namely mRNA, miRNA and protein data. The result was an object of type `MultiAssayExperiment`, whose main components were the following.

- `colData`, i.e. a dataframe containing the phenotipic characteristics of each sample. In our case, it mainly included clinical data.
- `ExperimentList`, containing the considered experiments.
- `sampleMap`, connecting all the considered elements.

Data preprocessing

As first preprocessing step, we extracted specific samples by exploiting the associated **barcode** (Figure 1). More precisely, we were only interested in **primary** solid tumors – in order to have a more homogeneus group of samples – and thus only retained samples with code *01* in the *Sample* part. Furthermore, we checked for the presence of **technical replicates** (repeated measurements from the same sample), to avoid overrepresentation. Those are recognizable by having the same patient identifier, i.e. the first 12 digits of the barcode in Figure 1, but weren't found in our dataset.

Later on, we removed **FFPE** (formalin-fixed, paraffin-embedded) samples, i.e. samples that were not frozen and whose DNA and RNA molecules were thus preserved in a less effective manner. After this phase, we retained only samples having all the three omics and saved them as matrices in a list; the matrices were transposed in order to have samples on rows and features on columns, while features having missing values were removed.

From the feature selection standpoint, we retained the first 100 features by **variance**: the assumption was that the higher the variance across samples, the higher the feature information content. Although frequently used in literature, this method does not remove redundant variables, does not consider features interactions and has the drawback of setting an arbitrary threshold (such as 100 in our case). The features were further preprocessed by performing **z-score** standardization.

Finally, we kept only the first 12 digits of the barcode, to identify the patient without keeping unused information.

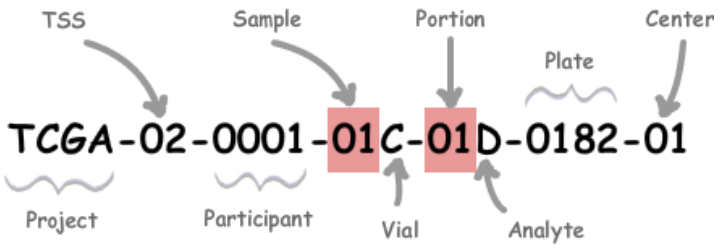

Figure 1: Structure of TCGA barcodes

2.2 Data integration

We used and compared three multiomics data integration methods, namely **Similarity Network Fusion (SNF)**, simple **Matrices Average** and **NEMO**.

Similarity Network Fusion

The first data integration approach was *Similarity Network Fusion (SNF)* [6]. The process starts by building a similarity matrix between samples for each data source s, leveraging the scaled exponential Euclidean distance:

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

having:

- $\rho(x_i, x_j)$ the Euclidean distance between patients x_i and x_j;
- μ a parameter;
- $\varepsilon_{i,j}$ is the scaling factor

$$\frac{\text{mean}(\rho(x_i, N_i)) + \text{mean}(\rho(x_j, N_j)) + \rho(x_i, x_j)}{3},$$

with N_i being the neighborhood of patient x_i, i.e. $N_i = \{x_k \mid x_k \in k\text{NN}(x_i) \cup \{x_i\}\}$.

This gives rise to a similarity matrix of size $m \times m$ for each data source, with m the number of patients. Finally, the exponential turns the Euclidean distance into a similarity metric.
A global similarity matrix $P^{(s)}$ is then computed, to capture relationships between patients:

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

Except for the main diagonal, elements of $P^{(s)}$ are the similarity divided by twice the sum over all the other matrix elements.
Finally, a local similarity matrix $L^{(s)}$ is computed to capture the network's local structure by considering only local similarities in each patient's neighborhood:

$$L^{(s)}(i,j) = \begin{cases} \dfrac{W^{(s)}(i,j)}{\sum_{k \in N_i} W^{(s)}(i,k)} & \text{if } j \in N_i \\ \\ 0 & \text{otherwise.} \end{cases}$$

Thus the similarity between two samples i and j is scaled by the total similarity of sample i with its neighborhood.
With S data sources, S different W, L, P matrices are built. Similarities should then be diffused through the Ps until convergence, but in practice a pre-defined number of steps is chosen (20 in our case).
Let ${P^{(s)}}_t$ be the matrix P for data source s at time t; we focus on the simplest case with only two data sources, so that $s \in \{1, 2\}$. The diffusion process can be described as:

$${P^{(1)}}_{t+1} = L^{(1)} \times {P^{(2)}}_t \times {L^{(1)}}^\top$$
$${P^{(2)}}_{t+1} = L^{(2)} \times {P^{(1)}}_t \times {L^{(2)}}^\top.$$

$P^{(1)}$ is updated using $L^{(1)}$, i.e. the local similarity matrix for the same data source, and $P^{(2)}$, i.e. the global similarity matrix for the other data source; vice versa for $P^{(2)}$.
The integrated matrix M is the average over each data source of all global similarity matrices:

$$M = \frac{1}{S} \sum_s P^{(s)}.$$

Matrices average

A trivial multi-omics data integration strategy was given by the computation of the average over all similarity matrices.
After building the matrix

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

for each data source s, we summed elements with the same row and column indices, then divided by the number of data sources:

$$M = \frac{1}{S} \sum_s W^{(s)}.$$

NEMO

NEighborhood based **M**ulti-**O**mics clustering is an algorithm for multi-omics clustering [14].
NEMO takes as input a set of matrices of m samples. Given S omics, let $X^{(s)}$ be the matrix for omic s: each $X^{(s)}$ has size $p_s \times m$, where p_s is the number of features for omic s.

Let x_{s_i} be column i of X_s, and N_{s_i} its k nearest neighbors within omic s using Euclidean distance.
For omic s, a $m \times m$ similarity matrix $W^{(s)}$ is built as such:

$$W^{(s)}(i,j) = \frac{1}{\sqrt{2\pi}\varepsilon_{i,j}} \exp\left(-\frac{\rho(x_i, x_j)^2}{\mu\, \varepsilon_{i,j}}\right)$$

We then define the relative similarity matrix $R^{(s)}$:

$$R^{(s)}(i,j) = \frac{W^{(s)}(i,j)}{\sum_{n \in N_{s_i}} W^{(s)}(i,n)} I(n \in N_{s_i}) + \frac{W^{(s)}(i,j)}{\sum_{n \in N_{s_j}} W^{(s)}(n,j)} I(n \in N_{s_j}),$$

where I is the indicator function. $R^{(s)}(i,j)$ measures the similarity between samples i and j w.r.t. i's k nearest neighbors and j's k nearest neighbors. The relative similarity can be compared more easily than the original similarity $W^{(s)}$, because different omics have different data distributions.
$R^{(s)}$ can be interpreted as a transition probability between samples, where the probability is proportional to their similarity. These transition distributions are the same used to describe random walk on graphs.
Finally, NEMO computes the $m \times m$ average relative similarity matrix M over omics:

$$M = \frac{1}{S} \sum_s R^{(s)}.$$

2.3 Clustering approaches

As clustering approaches, we used Partitioning Around Medoids (PAM) and Spectral Clustering.

Partition Around Medoids

Partitioning Around Medoids (PAM) [17] is a clustering algorithm that finds a fixed number of clusters k, given as input by the user and represented by their central points, or **medoids**.
Let O be the entire set of objects, and E the set of objects defined as medoids: then $U = O \setminus E$ is the set of non-selected objects.

We want to obtain a set of clusters such that the **average distances** of objects belonging to the cluster and the cluster representative is **minimized**.
First, PAM selects k initial objects to populate E, while the objects in U are assigned to the **closest representative** in E: the first representative object is

the one minimizing distance w.r.t. all the other objects, that is the most central point.

The other points i in U are selected as representatives if there are enough non-selected objects j closer to i than to already selected representatives in E; in other words, we are looking for the element with many non-representative objects close to it that is also further away than the already selected representatives. The previous steps are repeated until k medoids are found.
Next, the algorithm tries to improve the set E of selected representatives. For all pairs of representatives $i \in E$ and non-representatives $n \in U$ the following steps are performed.

- Elements i and n are **swapped**, so that n becomes a representative and i stops being one.
- The **contribution** K_{jin} of each object $j \in U \setminus \{n\}$ to the swap of i and n is computed, where n is removed because it is now a representative. Two situations may arise:
  - $d(j,i) > D_j$, with $d(j,i)$ the distance between i and j, and D_j the dissimilarity between j and the closest representative. Here $K_{jin} = \min\{d(j,n) - D_j, 0\}$
  - $d(j,i) = D_j$. Here $K_{jin} = \min\{d(j,n), C_j\} - D_j$, with C_j the dissimilarity between j and the second closest representative.
- The total results of the swap are computed as a sum over all contributions of non-representatives, that is $T_{in} = \sum_{j \in U} K_{jin}$.
- Pair (i,n) maximizing T_{in} is selected.
- If T_{in} is negative, the swap is performed and we obtain a better cluster, since we are minimizing distances inside it; D_j and E_j are then computed again a new pair of representative and non-representative is selected.

  If all T_{in} are strictly positive, the algorithm stops.

Spectral clustering

Once we have an integrated matrix M, the *Spectral Clustering Algorithm* [1] works as follows.

- Let G be the diagonal matrix whose element in position (i,i) is the sum of the i-th row of M, then build matrix $L = G^{-1/2} M D^{-1/2}$.
- With $x_1, \ldots, x_k$ the k largest eigenvectors of L, build matrix $X = [x_1 \ldots x_k] \in \mathbb{R}^{m \times k}$ by stacking the eigenvectors in columns.
- Normalize each row of X to have unit length, forming matrix Y.
- Interpreting each row of Y as a point in $\mathbb{R}^k$, cluster them into k clusters, for instance using K-means.

- Assign the original point x_i to cluster j if and only if row i of matrix Y was assigned to cluster j.

2.4 Clusters evaluation

The resulting clusters were evaluated against the integrative ones obtained by [7] using iCluster [2]: the association between the three subtypes and the samples was reached through the `Subtype Integrative` column, after excluding samples that were not present both in our dataset and in the subtypes. For the same reason, the number of clusters was set to three in all algorithms.

The comparison exploited three of the most famous cluster indices, present in the `mclustcomp` R package and listed below.

- **Rand index (RI)**: based on *counting pairs* of objects that are in the same cluster in both clusterings C_1 (iCluster) and C_2 (ours). Thus, with regards to all the possible pairs, we have

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

  where n_{11} is the number of objects pairs that are in the same clusters both in C_1 and C_2, while n_{00} is the number of pairs that are in different clusters both in C_1 and C_2. Rand index ranges from 0 to 1, where 1 indicates identical clusterings and 0 completely different clusterings.

- **Adjusted Rand Index (ARI)**: corrects RI for chance, assuming a generalized hypergeometric distribution as null hypothesis. It is computed as the normalized difference of the RI and its expected value under the null hypothesis. ARI ranges from 0 to 1 too, where 1 indicates identical clusterings and zero independent clusterings.

- **Normalized Mutual Information (NMI)**: it quantifies how much we can reduce uncertainty about the cluster of an element when we already know its cluster in another clustering. We thus have

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

  where $P(i,j) = \frac{|C_{1i} \cap C_{2j}|}{n}$, i.e. the probability that $i \in C_1$ and $j \in C_2$.

  To foster interpretability – since Mutual information is not upper bounded – we use a normalized version, precisely:

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

  NMI ranges between 0 and 1, where maximum NMI is reached if $C_1 = C_2$.

3 Results

Table 1 shows the performance scores obtained for the different **combinations** of **integration** and **clustering** methods, according to the three indices described above.

	RI	ARI	NMI
mRNA + PAM	0.56	0.03	0.04
miRNA + PAM	0.58	0.06	0.06
Protein + PAM	0.54	0.00	0.01
Average + PAM	0.57	0.04	0.07
SNF + PAM	0.60	0.12	0.12
NEMO + PAM	0.54	0.03	0.05
NEMO + Spectral	0.38	0.01	0.07
SNF + Spectral	0.60	0.12	0.12

Table 1: Compared results from the implemented approaches

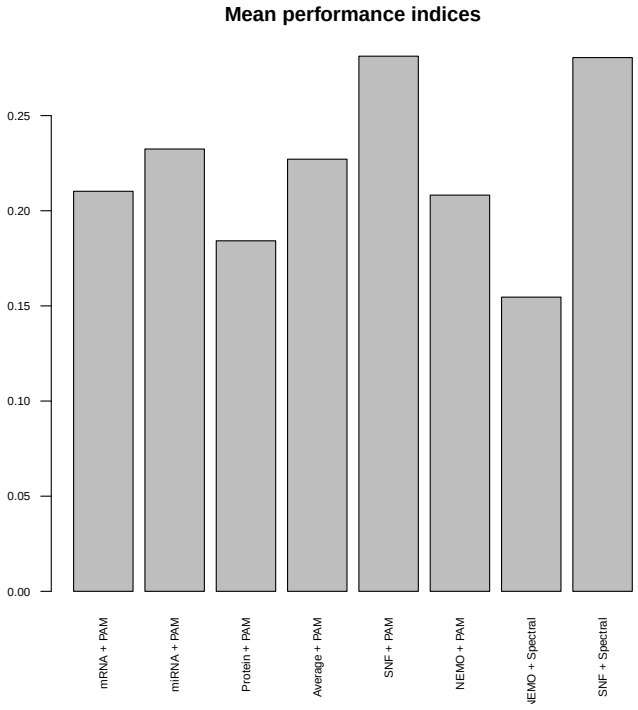

Figure 2: An aggregated view over performance indices

One first thing that can be noticed is that the **RI** index is much more **opti-**

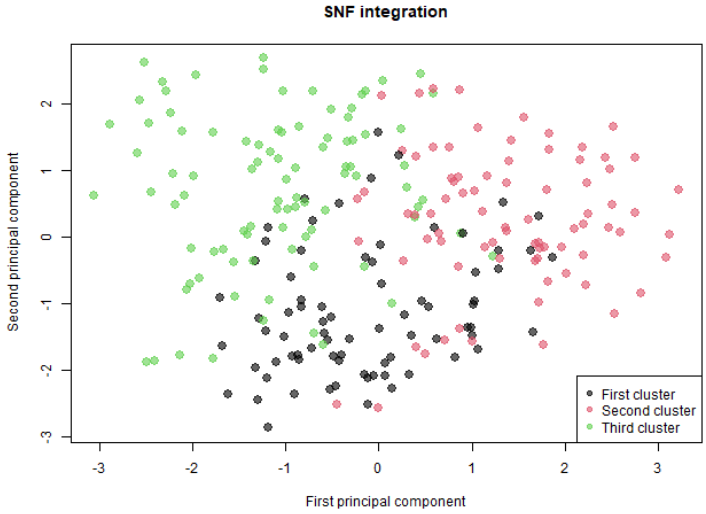

Figure 3: Visualization of SNF + PAM clustering, after applying PCA

mistic than the other ones, which are more likely to be correcting for chance; this highlights the importance of considering different measures to evaluate results.

The overall **diversity** between our clusters and the *iCluster* ones could be motivated by the different ways in which the methods operate and thus the distinct properties they capture: while *iCluster* relies on a **Bayesian** approach, SNF and NEMO are **network**-based and **similarity**-based respectively.

It is, however, interesting to notice that using **single omics** is not changing much the indices with respect to using NEMO, despite NEMO and *iCluster* being both **integrative** methods: on the contrary, **protein** omics shows a higher value for all indices not only when compared to other single omics and their average, but also with respect to NEMO.

NEMO results might also have been influenced by the fact that – although one of its main advantages is that it can be applied on partial data [14] – we were not able to use it in presence of null values, and thus had to remove them.

Figure 2 shows a plot of the results averaged across indices: it can be noticed that SNF always has the best results, almost independently on the clustering method used.

By having a look at the PCA graphs, in fact, we can visualize that SNF (Figure 3) makes clusters more separable in the feature space than NEMO (Figure 4). Additionally – integration approach (SNF) being equal – Spectral Clustering (Figure 5) seems to better discriminate between clusters in the feature space when compared to PAM (Figure 3).

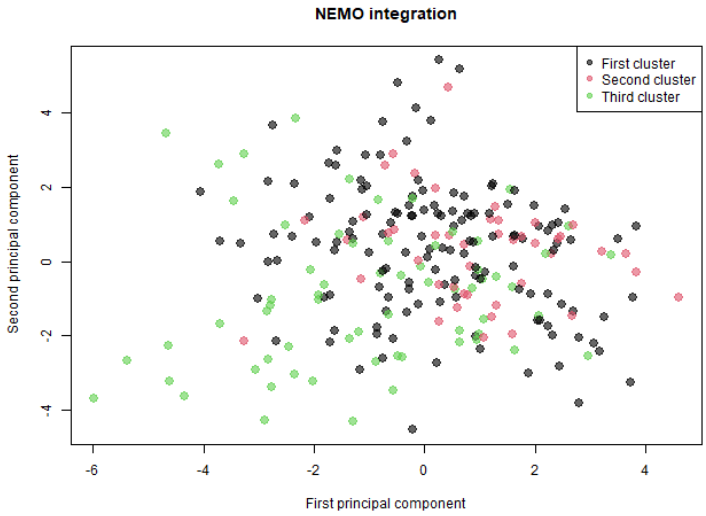

Figure 4: Visualization of NEMO + PAM clustering, after applying PCA

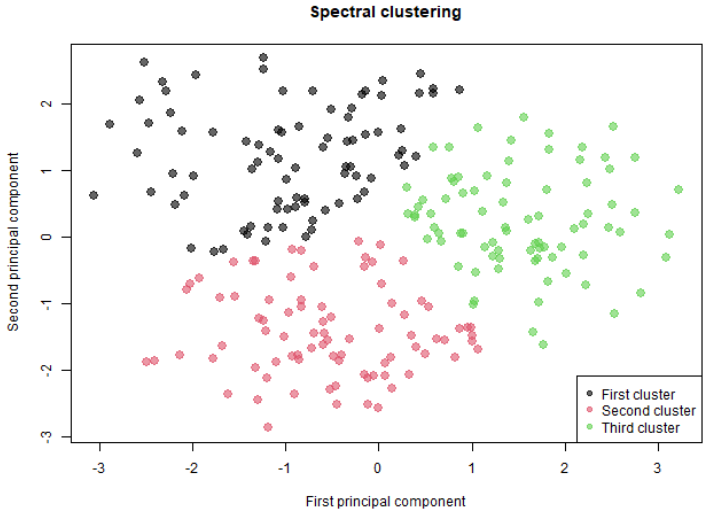

Figure 5: Visualization of SNF + Spectral clustering, after applying PCA

References

[1] Andrew Ng, Michael Jordan, and Yair Weiss. "On Spectral Clustering: Analysis and an algorithm". In: *Advances in Neural Information Processing Systems*. Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press, 2001. URL: https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf.

[2] Ronglai Shen, Adam B. Olshen, and Marc Ladanyi. "Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis". In: *Bioinformatics* 25.22 (Sept. 2009), pp. 2906–2912. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp543. eprint: https://academic.oup.com/bioinformatics/article-pdf/25/22/2906/48997731/bioinformatics_25_22_2906.pdf. URL: https://doi.org/10.1093/bioinformatics/btp543.

[3] Charles J. Vaske et al. "Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using PARADIGM". en. In: *Bioinformatics* 26.12 (June 2010), pp. i237–45.

[4] Eric F. Lock et al. "Joint and Individual Variation Explained (jive) for integrated analysis of multiple data types". en. In: *Ann. Appl. Stat.* 7.1 (Mar. 2013), pp. 523–542.

[5] Qianxing Mo et al. "Pattern discovery and cancer gene identification in integrated cancer genomic data". en. In: *Proc. Natl. Acad. Sci. U. S. A.* 110.11 (Mar. 2013), pp. 4245–4250.

[6] Bo Wang et al. "Similarity network fusion for aggregating data types on a genomic scale". en. In: *Nat. Methods* 11.3 (Mar. 2014), pp. 333–337.

[7] Adam Abeshouse et al. "The Molecular Taxonomy of Primary Prostate Cancer". In: *Cell* 163.4 (Nov. 2015), pp. 1011–1025. DOI: 10.1016/j.cell.2015.10.025. URL: https://doi.org/10.1016/j.cell.2015.10.025.

[8] Suchi Saria and Anna Goldenberg. "Subtyping: What it is and its role in precision medicine". In: *IEEE Intell. Syst.* 30.4 (July 2015), pp. 70–75.

[9] Christopher Sweeney et al. "Disease-free survival (DFS) as a surrogate for overall survival (OS) in localized prostate cancer (CaP)". en. In: *J. Clin. Oncol.* 34.15_suppl (May 2016), pp. 5023–5023.

[10] Yehudit Hasin, Marcus Seldin, and Aldons Lusis. "Multi-omics approaches to disease". In: *Genome biology* 18.1 (2017), pp. 1–15.

[11] Florian Rohart et al. "mixOmics: An R package for 'omics feature selection and multiple data integration". In: *PLOS Computational Biology* 13.11 (Nov. 2017), pp. 1–19. DOI: 10.1371/journal.pcbi.1005752. URL: https://doi.org/10.1371/journal.pcbi.1005752.

[12] Qianqian Shi et al. "Pattern fusion analysis by adaptive alignment of multiple heterogeneous omics data". In: *Bioinformatics* 33.17 (Apr. 2017), pp. 2706–2714. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx176. eprint: https://academic.oup.com/bioinformatics/article-pdf/33/17/2706/49040717/bioinformatics_33_17_2706.pdf. URL: https://doi.org/10.1093/bioinformatics/btx176.

[13] Xiang-Tian Yu and Tao Zeng. "Integrative analysis of omics big data". In: *Computational Systems Biology: Methods and Protocols* (2018), pp. 109–135.

[14] Nimrod Rappoport and Ron Shamir. "NEMO: cancer subtyping by integration of partial multi-omic data". In: *Bioinformatics* 35.18 (Jan. 2019), pp. 3348–3356. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz058. eprint: https://academic.oup.com/bioinformatics/article-pdf/35/18/3348/48975530/bioinformatics_35_18_3348.pdf. URL: https://doi.org/10.1093/bioinformatics/btz058.

[15] Indhupriya Subramanian et al. "Multi-omics data integration, interpretation, and its application". en. In: *Bioinform. Biol. Insights* 14 (Jan. 2020), p. 1177932219899051.

[16] Otília Menyhárt and Balázs Győrffy. "Multi-omics approaches in cancer research with applications in tumor subtyping, prognosis, and diagnosis". In: *Computational and Structural Biotechnology Journal* 19 (2021), pp. 949–960. ISSN: 2001-0370. DOI: https://doi.org/10.1016/j.csbj.2021.01.009. URL: https://www.sciencedirect.com/science/article/pii/S2001037021000131.

[17] "Partitioning Around Medoids (Program PAM)". In: *Finding Groups in Data*. John Wiley & Sons, Inc., pp. 68–125. DOI: 10.1002/9780470316801.ch2. URL: https://doi.org/10.1002/9780470316801.ch2.

[18] *The Cancer Genome Atlas*. https://www.cancer.gov/ccg/research/genome-sequencing/tcga. Accessed: 06 19, 2023.