Bioinformatics exam project report: Disease subtype discovery on prostate adenocarcinoma using multi-omics data integration

Davide Varricchio

December 2023

Contents

1 Introduction

The identification of disease subtype has the goal to find homogeneous groups of patients that have similar clinical and/or molecular characteristics. Finding these groups may be useful to personalize the therapy of the patient or to predict their prognosis, because different subtypes of disease need to be treated in different ways: for example, cancer is highly heterogeneous, with wide molecular differences even within the same tissue of origin. This is an integral part of what is known as precision medicine, an innovative approach to tailor disease prevention and treatment that takes into account differences in people's genes, environments, and lifestyles. In recent years, multi-omics data integration has emerged as a powerful approach for disease subtype discovery: by combining multiple layers of biological information (e.g. genome, proteome, transcriptome) a more comprehensive view of the heterogeneity of the disease can be obtained. The integration of different omics data is an open problem in scientific literature, initially simple methods like concatenation or averaging were proposed, but ignored complex relationships between data types. More advanced methods aim to model nonlinear relationships. These include graph-based integration like Similarity Network Fusion (SNF) [5] and NEighborhood based Multi-Omics (NEMO) [3], or joint latent variable models like integrative Clustering (iCluster) [4]. The aim of this project is to perform disease subtype discovery on primary prostate adenocarcinoma. The considered subtypes are those identified in a work performed by The Cancer Genome Atlas Research Network [1], where they used iCluster on multi-omics data (somatic copy-number alterations, methylation, mRNA, microRNA, and protein levels) and discovered three disease subtypes. In this work SNF is used to integrate multi-omics data (miRNA,RNA,Protein), and clustering with the Partitioning Around Medoid (PAM) algorithm [2] is performed. Then, the obtained clusters are compared with the iCluster ones, using as base lines PAM clusters computed on single data sources and on a merging acquired by averaging them.

2 Methods

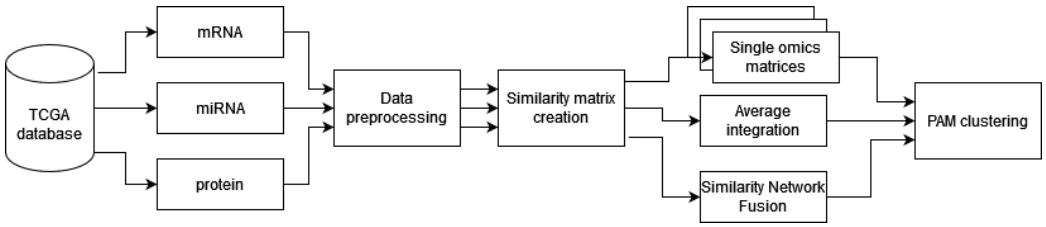

Figure 1: Block diagram of the methodology

2.1 Data

The data about prostate adenocarcinoma is obtained from The Cancer Genome Atlas Program (TCGA), a project that collected more than 11,000 cases across 33 tumor types, from multiple biological data sources. In particular, the following data sources have been selected:

- mRNA expression obtained via RNA sequencing
- miRNA expression obtained via RNA sequencing
- protein expression obtained via Reverse-Phase Protein Array

After obtaining the data, some pre-processing steps are needed. In the TCGA dataset, each sample is identified by a barcode, where the first 12 characters identify a specific individual, while the remaining part gives information about the type of sample (i.e. primary, metastatic, solid, blood derived, etc.), the type of genomic material extracted (i.e. DNA, RNA) and other information related to technical replicates (i.e. repeated measurements from the same sample). the barcode is used to retain only Primary Solid Tumors and to check and eventually remove technical replicates. The formalin-fixed, paraffin-embedded (FFPE) samples are removed, so that only the ones preserved by freezing are retained. This is done because DNA and RNA molecules are preserved better if the tissue is frozen. Not all samples have all the considered omics, because of this, the records that miss some are deleted from the dataset. Since it is common for biological dataset to have features as rows and samples as columns, the 3 matrices for every data source are transposed, so that it is easier to work in the machine learning environment. Then, the following pre-processing steps are applied for every data source matrix:

- Remove features having missing values. In this case it is easier to remove features instead of performing imputation, since only few features in the proteomics data have missing values.
- Remove the features with near zero variance. This eliminates features that are unlikely to distinguish between subtypes, with the assuption that features with more variance across samples bring more information. For a feature to be flagged, first the frequency of the most prevalent value over the second most frequent value must be above 95/5. Secondly, the "percent of unique values" i.e. the number of unique values divided by the total number of samples (times 100), must also be below 10.
- Retain only the top 100 features with higher variance, assuming that they carry most of the information.
- Perform feature standardization using z-score
- Clean barcodes to retain only the first part specific for each individual

It needs to be noted about the filtering by variance steps that, while it is a fast and commonly used technique to reduce dimensionality, it has some drawbacks:

- it is univariate, thus does not considers interactions among features.
- it is not able to remove redundant variables.
- a threshold needs to be identified for feature selection (top 100 features), but it is always an arbitrary choice.

The disease subtype data [1] is also obtained from TCGA. In particular, the column "Subtype_Integrative" is the one containing the iCluster molecular subtypes. This dataset is pre-processed so that it contains only records of primary solid prostate cancer that do not have missing values in the column of interest. Finally, only samples in common between the omics and the subtype datasets are retained.

2.2 Multi-omics integration

To integrate the multi-omics data, the SNF [5] method is used. The first step, is a construction of a similarity matrix among samples for each data

source. The similarity measure exploited is the scaled exponential euclidean distance:

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

where:

- $\rho(x_i, x_j)$ is the Euclidean distance between patients x_i and x_j
- μ is a parameter. The default 0.5 library value is used.
- $\varepsilon_{i,j}$ is a scaling factor: $\varepsilon_{i,j} = \frac{mean(\rho(x_i,N_i))+mean(\rho(x_j,N_j))+\rho(x_i,x_j)}{3}$, where $mean(\rho(x_i, N_i))$ is the average value of the distances between x_i and each of its neighbours. As neighbour size, the library standard value of 20 is used.

Then, other two matrices are derived from $W^{(s)}(i,j)$. One is a "global" similarity matrix $P^{(s)}$ which is needed to capture the overall relationships between patients:

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

For this equation the property $\sum_j P(i,j) = 1$ holds. The other one is a "local" similarity matrix $S^{(s)}$, that captures the local structure of the network because considers only local similarities in the neighborhood of each individual, setting to zero all the others:

$$S^{(s)}(i,j) = \begin{cases} \frac{W^{(s)}(i,j)}{\sum_{k\in N_i} W^{(s)}(i,k)} & \text{, if } j \in N_i \\ 0 & \text{, otherwise} \end{cases} \tag{3}$$

where $N_i = \{x_k | x_k \in kNN(x_i) \cup \{x_i\}\}$. Given s data modalities, s different W, S and P matrices are constructed and an iterative process is applied where similarities are diffused through the Ps until convergence, that is, until all the matrices P become similar:

$$P^{(v)} = S^{(v)} \times (\frac{\sum_{k\neq v} P^{(k)}}{s-1}) \times (S^{(v)})^T, v = 1, 2..., s \tag{4}$$

In other words $P^{(v)}$ is updated by using $S^{(v)}$ from the same data source but $P^{(k)}$ from different views. The number of iterations used is 20, as well as the neighbour size. The final integrated matrix $P^{(c)}$ is computed by averaging:

$$P^{(c)} = \frac{1}{s}\sum_{k=1}^{s} P^{(k)} \tag{5}$$

As a baseline, another simple integrated matrix is computed by averaging the similarity matrices:

$$W^{(c)}(i,j) = \frac{\sum_{k=1}^{s} W^{(k)}(i,j)}{s} \tag{6}$$

2.3 Clustering

Once a single integrated matrix is available, a clustering algorithm can be applied to try to identify disease subtypes. The algorithm used is PAM [2], which finds a fixed number of clusters k (given as input by the user) that are represented by their central points called medoids. The entire set of objects is O and the set of objects that are tentatively defined as medoids is S, so $U = O - S$ is the set of unselected objects. The final goal is to obtain a set of clusters where the average distances of objects belonging to the cluster and the cluster representative is minimized (equivalently the sum of the distances can be minimized). In particular, the algorithm has two phases:

- BUILD PHASE: the goal is to select k initial objects to populate the set of selected objects S. Then, the other objects in U are assigned to the closest representative in S. The first object in S is the one that has minimal distance with all the other objects, thus the most central data point. The other points i in U are evaluated to be selected as representatives and chosen if they have a high number of unselected objects j that are closer to i than to already selected representatives belonging to S. These steps are performed until a number of selected medoids k is reached.

- SWAP PHASE: this phase is intended to improve the set of selected representatives. For each pair of representative $i \in S$ and non representative $h \in U$:

  - We swap i and h, as that h is a representative and i is not. Compute the contribution K_{jih} of each object $j \in U - \{h\}$ to the swap of i and h. We can have two main situations:

    * $d(j,i) > D_j$, where D_j is the dissimilarity between j and the closest object in S. Then, $K_{jih} = min\{d(j,h) - D_j, 0\}$.
    * $d(j,i) = D_j$. 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.

  - Compute the total results of the swap as $T_{ih} = \sum\{K_{jih} | j \in U\}$.

- Select the pair i, h that minimizes T_{ih}.
- If $T_{ih} < 0$ the swap is performed, D_j and E_j are recomputed and we return at the first step of the SWAP phase. Otherwise, the algorithm stops if all $T_{ih} > 0$.

Since PAM requires to operate with dissimilarities, the similarity matrices are converted to distance ones, after performing a max-min normalization:

$$W_{dist} = 1 - \frac{W - min(W)}{max(W) - min(W)} \tag{7}$$

Then, the algorithm is applied with $k = 3$ (like the 3 iCluster subtypes [1]) on the following matrices:

- SNF integrated matrix.
- Average integrated matrix.
- 3 Matrices of the 3 single data sources.

2.4 Evaluation metrics

To compare the obtained clusters with the subtypes already available, 3 popular metrics are computed:

- Rand Index (RI): this index is one of the measures based on "counting pairs" of objects that are in the same cluster in both clusterings C_1 and C_2. It just counts the number of objects pairs that are in the same clusters both in C_1 and C_2 (defined by n_{11}) and the number of pairs that are in different clusters both in C_1 and C_2 (defined by n_{00}) w.r.t. all the possible pairs:

  $$R(C_1, C_2) = \frac{2(n_{11} + n_{00})}{n(n-1)} \tag{8}$$

  Rand index ranges from 0 to 1, where 1 indicates identical clusterings and 0 completely different clusterings.

- Adjusted Rand Index (ARI): If we consider two random partitions of a dataset, the Rand index does not assume a constant value (e.g. 0). Thus, the ARI is the RI corrected-for-chance, that assumes a generalized hypergeometric distribution as null hypothesis. ARI is the (normalized) difference of the RI and its expected value under the null hypothesis. ARI ranges from -0.5 to 1, where 1 indicates identical clusterings and near zero values independent clusterings.

- Normalized Mutual Information (NMI): the mutual information is a way to quantify how much we can reduce uncertainty about the cluster of an element when we already know its cluster in another clustering:

$$MI(C_1, C_2) = \sum_{i=1}^{k} \sum_{j=1}^{l} P(i,j) log_2 \frac{P(i,j)}{P(i)P(j)} \tag{9}$$

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 $C_j \in C_2$. Mutual information is not upper bounded, thus its interpretability is difficult. A normalized version of the mutual information is:

$$NMI(C_1, C_2) = \frac{MI(C_1, C_2)}{\sqrt{H(C_1)H(C_2)}} \tag{10}$$

where $H(C_1)$ and $H(C_2)$ are the entropies associated to clusterings C_1 and C_2. NMI ranges between 0 and 1, where maximum NMI is reached if $C_1 = C_2$.

3 Results

Table 1: Evaluation metrics for each integration approach

Approach	AdjRand	NMI	Rand
SNF integration	0.1795	0.1567	0.6317
Avg integration	0.0237	0.0403	0.5598
miRNA	0.0247	0.0276	0.5424
mRNA	0.0377	0.0532	0.5575
proteins	0.0079	0.0197	0.5524

Among the single data sources, mRNA seems to be the one that performs better, while proteins perform the worst. Anyway, they all have similar Rand index, which gives a more optimistic score. Thus to have a fair interpretation of the results is always better to consider different measures. The Average method yields similar scores to the single omics, which can be expected since it is a naive integration method, that doesn't take into account complex non linear relationship between the data. The SNF integration approach on the other hand, is able to do so, and it has shown to dominate every other method in all the considere metrics. This is better visualized in the following bar plot:

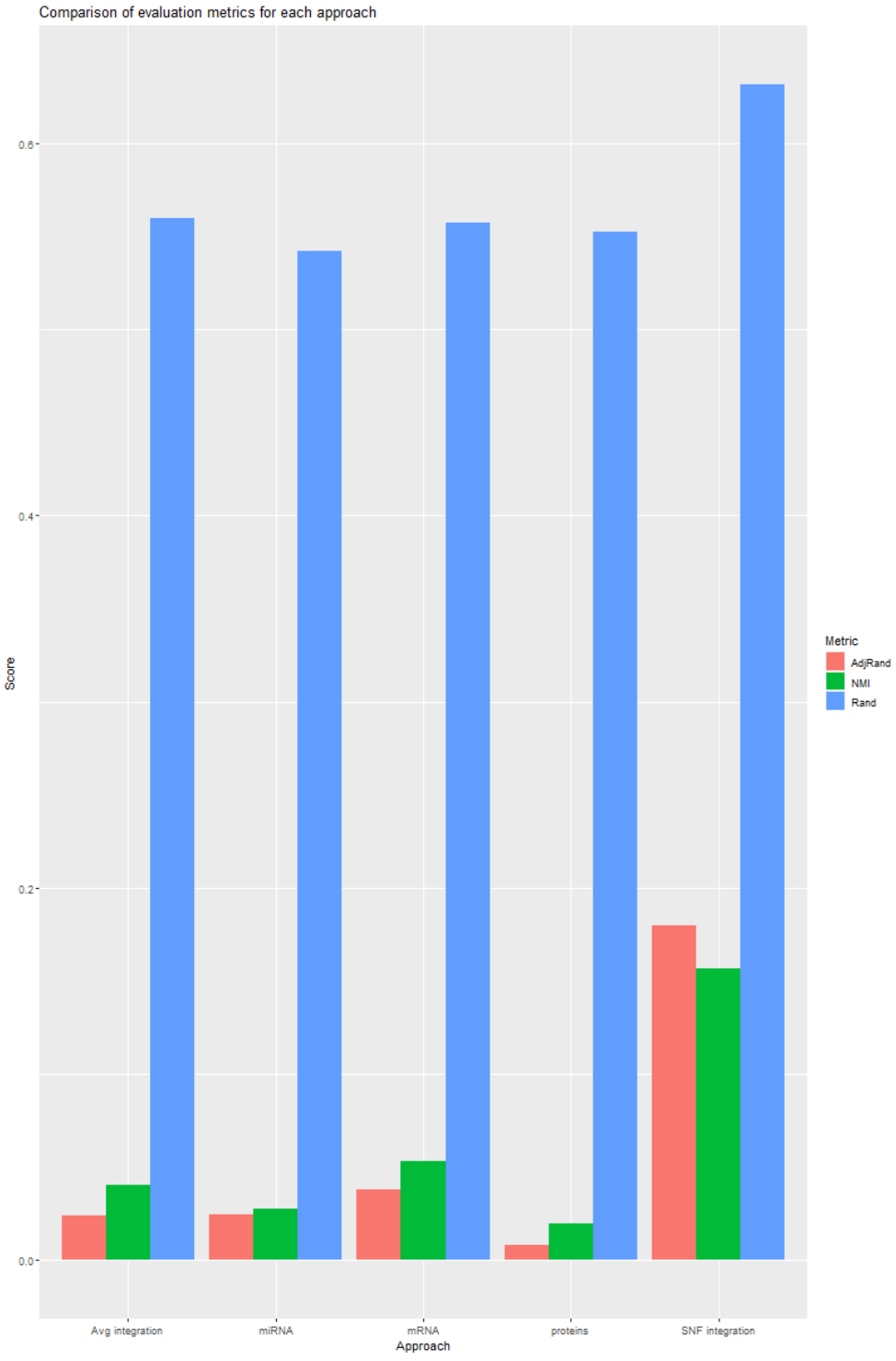

Figure 2: Grouped bar plot of the results of each different approach

Even though the SNF metrics are better, they indicate a limited overlap with the iCluster subtypes. This is not a surprise, as in this work miRNA, mRNA and protein expression data were used, while the subtypes already available were obtained using also somatic copy-number alterations and methylation omics. Nonetheless, other important arbitrary factors contribute to these results:

- Integration algorithm and their parameters, like the neighbour size in SNF.
- Clustering algorithm used.
- Crucial pre-processing steps, like the filtering by variance ones, which limits are explained in section 2.1

4 Session Info

This section contains the session info of the R environment used:

- R version 4.3.2 (2023-10-31 ucrt), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=Italian_Italy.utf8`, `LC_CTYPE=Italian_Italy.utf8`, `LC_MONETARY=Italian_Italy.utf8`, `LC_NUMERIC=C`, `LC_TIME=Italian_Italy.utf8`
- Time zone: `Europe/Rome`
- TZcode source: `internal`
- Running under: `Windows 10 x64 (build 19043)`
- Matrix products: default
- Base packages: base, datasets, graphics, grDevices, methods, stats, stats4, utils
- Other packages: Biobase 2.62.0, BiocGenerics 0.48.1, caret 6.0-94, cluster 2.1.4, curatedTCGAData 1.24.0, GenomeInfoDb 1.38.1, GenomicRanges 1.54.1, ggplot2 3.4.4, IRanges 2.36.0, lattice 0.22-5, MatrixGenerics 1.14.0, matrixStats 1.1.0, mclust 6.0.1, mclustcomp 0.3.3, MultiAssayExperiment 1.28.0, NetPreProc 1.2, S4Vectors 0.40.2, SNFtool 2.3.1, SummarizedExperiment 1.32.0, TCGAbiolinks 2.30.0, TCGAutils 1.22.0

- Loaded via a namespace (and not attached): abind 1.4-5, alluvial 0.1-2, AnnotationDbi 1.64.1, AnnotationHub 3.10.0, BiocBaseUtils 1.4.0, BiocFileCache 2.10.1, BiocIO 1.12.0, BiocManager 1.30.22, BiocParallel 1.36.0, BiocVersion 3.18.1, biomaRt 2.58.0, Biostrings 2.70.1, bit 4.0.5, bit64 4.0.5, bitops 1.0-7, blob 1.2.4, cachem 1.0.8, class 7.3-22, cli 3.6.1, codetools 0.2-19, colorspace 2.1-0, compiler 4.3.2, crayon 1.5.2, curl 5.1.0, data.table 1.14.8, DBI 1.1.3, dbplyr 2.4.0, DelayedArray 0.28.0, digest 0.6.33, downloader 0.4, dplyr 1.1.4, ellipsis 0.3.2, ExperimentHub 2.10.0, ExPosition 2.8.23, fansi 1.0.5, farver 2.1.1, fastmap 1.1.1, filelock 1.0.2, foreach 1.5.2, future 1.33.0, future.apply 1.11.0, generics 0.1.3, GenomeInfoDbData 1.2.11, GenomicAlignments 1.38.0, GenomicDataCommons 1.26.0, GenomicFeatures 1.54.1, globals 0.16.2, glue 1.6.2, gower 1.0.1, graph 1.80.0, grid 4.3.2, gtable 0.3.4, hardhat 1.3.0, hms 1.1.3, htmltools 0.5.7, httpuv 1.6.13, httr 1.4.7, interactiveDisplayBase 1.40.0, ipred 0.9-14, iterators 1.0.14, jsonlite 1.8.8, KEGGREST 1.42.0, knitr 1.45, labeling 0.4.3, later 1.3.2, lava 1.7.3, lifecycle 1.0.4, listenv 0.9.0, lubridate 1.9.3, magrittr 2.0.3, MASS 7.3-60, Matrix 1.6-4, memoise 2.0.1, mime 0.12, ModelMetrics 1.2.2.2, munsell 0.5.0, nlme 3.1-164, nnet 7.3-19, parallel 4.3.2, parallelly 1.36.0, pillar 1.9.0, pkgconfig 2.0.3, plyr 1.8.9, png 0.1-8, prettyGraphs 2.1.6, prettyunits 1.2.0, pROC 1.18.5, prodlim 2023.08.28, progress 1.2.3, promises 1.2.1, purrr 1.0.2, R6 2.5.1, rappdirs 0.3.3, rbibutils 2.2.16, Rcpp 1.0.11, RCurl 1.98-1.13, Rdpack 2.6, readr 2.1.4, recipes 1.0.8, reshape2 1.4.4, restfulr 0.0.15, rjson 0.2.21, rlang 1.1.2, rpart 4.1.23, Rsamtools 2.18.0, RSQLite 2.3.3, rtracklayer 1.62.0, rvest 1.0.3, S4Arrays 1.2.0, scales 1.3.0, shiny 1.8.0, SparseArray 1.2.2, splines 4.3.2, stringi 1.8.2, stringr 1.5.1, survival 3.5-7, TCGAbiolinksGUI.data 1.22.0, tibble 3.2.1, tidyr 1.3.0, tidyselect 1.2.0, timechange 0.2.0, timeDate 4022.108, tools 4.3.2, tzdb 0.4.0, utf8 1.2.4, vctrs 0.6.5, withr 2.5.2, xfun 0.41, XML 3.99-0.16, xml2 1.3.6, xtable 1.8-4, XVector 0.42.0, yaml 2.3.7, zlibbioc 1.48.0

References

[1] Adam Abeshouse et al. "The molecular taxonomy of primary prostate cancer". In: *Cell* 163.4 (2015), pp. 1011–1025.

[2] "Partitioning Around Medoids (Program PAM)". In: *Finding Groups in Data*. John Wiley & Sons, Ltd, 1990. Chap. 2, pp. 68–125. ISBN: 9780470316801. DOI: https://doi.org/10.1002/9780470316801.ch2. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470316801.ch2.

[3] Nimrod Rappoport and Ron Shamir. "NEMO: cancer subtyping by integration of partial multi-omic data". In: *Bioinformatics* 35.18 (2019), pp. 3348–3356.

[4] 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. URL: https://doi.org/10.1093/bioinformatics/btp543.

[5] Bo Wang et al. "Similarity network fusion for aggregating data types on a genomic scale". In: *Nature methods* 11.3 (2014), pp. 333–337.