Algorithms for Big Data Analysis in Biology and Medicine **Fall, 2017-8**

Lecture 9: Multi Omics Clustering December 19, 2017

Lecturer: Nimrod Rappoport *Scribe: Shahar Segal*

9.1 Introduction to Multi-Omics

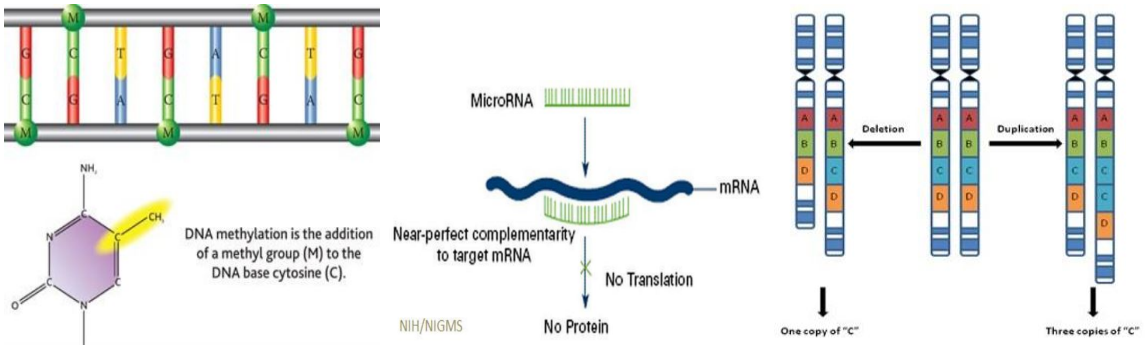
9.1.1 Omics

Omic is a term used to describe a field of study in biology that utilizes a certain type of biological data (e.g. genomics is the study of genome), while multi omics is the usage of several types of omics data.
During the course we've mainly discussed mRNA data. In the introduction lecture other types of biological data were shown, protein and DNA data (referred to as proteomics and genomics). There are many other omics, of which we'll present additional three: methylation, microRNA and copy number variation.

- Methylation is the process of adding the chemical group methyl to a cytosine DNA base (C). Methylation on a gene promoter usually represses transcription of that gene (decreases expression). Methylation differ between cells and changes during our life time, which was shown to be useful for age prediction [1], making methylation a relevant omic for other scientific fields, such as forensics.

- MicroRNA, abbreviated miRNA, are small RNA molecules that repress mRNA translation by attaching to the mRNA and preventing it from being translated, thus decreasing the protein levels of a gene, even if there's a high concentration of mRNA from which it can be translated.

- Copy number variations is a phenomenon in which genes are duplicated or deleted from the genome. Almost all genes have two copies in a healthy cell, one on each chromosome. It is common in cancer to have different copy number for different genes due to the instability of its replication process.



(a) Source:[2] (b) Source:[3] (c) Source:[4]

Figure 9.1: **(a)** DNA methylation. the methyl group (-CH3) is attached to cytosine (C). **(b)** MicroRNA. the miRNA binds to a target mRNA suppressing its translation. **(c)** Copy number variation. To the left a deletion event, changing gene C's copy number to 1. To the right a duplication event, changing gene C's copy number to 3.

9.1.2 Multi omics clustering for cancer subtyping

Cancers are known to be heterogeneous. That is, there are subtypes of cancer even within the same tissue. It is also known that multiple types of biological entities have a role in cancer prognosis. This makes multi-omics based clustering a viable approach in cancer study. Because of that, several organizations in recent years have been collecting multiple omics data on cancer using high throughput methods. One notable is *TCGA*.

TCGA **T**he **C**ancer **G**enome **A**tlas is a project aiming to improve the prevention, diagnosis and treatment of cancer. TCGA collects and analyses multi-omics data from cancer patients using high-throughput techniques and currently possesses genomic data of over 11,000 patients and more than 30 types of tumors (that's more than 2.5 petabytes of data).

Multi omics raises new challenges due to the variety in data forms, making data refinement and integration non-trivial. e.g. mutations can be binary, copy number variation uses discrete values, while gene expression is continuous and DNA methylation uses beta value, a ratio between methylated and unmethylated loci in each methylation site, where 0 is completely unmethylated and 1 is fully methylated.

9.1.3 Multi omics data approaches

To handle the challenges mentioned in section 9.1.2, different approaches were developed for integrating the multiple data types. These approaches can be catalogued by the timing of integration (when to integrate the multi-omic data) and by specificity of the method to its omics data types (how specific is the method and can it be generalized to other omics).

- Integration Stage:
  - Early integration - Concatenate the matrices and perform the clustering on the new matrix. Pros: simple, and allows us to use general clustering algorithm. Cons: the merged matrix is of high dimension and we disregard the difference in data type per omic.
  - Late integration - Perform the clustering on each omic separately and integrate the clustering results. Pros: simple, allows us to use general clustering algorithms. Cons: does not address the relations between different features from different omics.
  - Intermediate integration - Uses all omics to build the model, but unlike the early approach, regards them as different views of the clusters. Pros: Takes advantage of relations between omics. Cons: Complex.
- Data type specificity:
  - Generic - Offers full support for any omic data type. Pros: Highly flexible, easy to work with, can be used outside the context of biology and multi-omics. Cons: Loss of biological knowledge. By being general we might lose knowledge about the biological role of each omic and the relations between them; Different data types might prove challenging to integrate e.g. continuous and discrete data.
  - Omic specific - Only support omics the algorithm was explicitly designed for. Pros: Takes full advantage of prior biological knowledge. Cons: Inflexible, might be outdated if new knowledge about the omics emerges.
  - Omic specific feature representation - Instead of being omic specific, convert multi-omic data into a representation which is generic but incorporate the relations and prior knowledge. For example, represent genomic data using the average value for each pathway. Pros: Take advantage of relations between omics, somewhat flexible, might lower the dimension. Cons: Complex, loses some biological knowledge.

9.1.4 Comparing Clusterings

In cancer subtypes there is no "gold standard" to compare your clustering method to. Clustering can be compared via synthetic data, data we generate by ourselves with a known solution. Synthetic data can easily measure how well the clustering was able to match the original data, but since it's synthetic, there's no guarantee it has any meaning in cancer subtypes. Another approach is to test the clustering by prognosis or other clinical features, such as performing survival analysis between clusters to see if the clusters have significant impact on the survival rate. There are also general criteria such as homogeneity, separation, *silhouette score*, etc.

9.1.5 Silhouette Score

Silhouette score is a method of interpretation and validation of consistency within clusters of data. The score measures how similar an object is to its own cluster (homogeneity) compared to other clusters (separation). It ranges from -1 to 1 where the higher the value the more the object is well matched to its own cluster and poorly matches the neighbouring clusters. Formally, let i be a single sample. Let a(i) denote the average distance of i to points within its cluster, and b(i) the average distance of i to points within the closest cluster it does not belong to. The Silhouette score for i:

$$s(i) = \frac{b(i) - a(i)}{max(b(i), a(i))}$$

The silhouette score for the data is the average score across all samples. That is, $\frac{1}{n}\sum_{i=1}^{n} s(i)$.

9.2 COCA

9.2.1 Cluster of cluster assignments

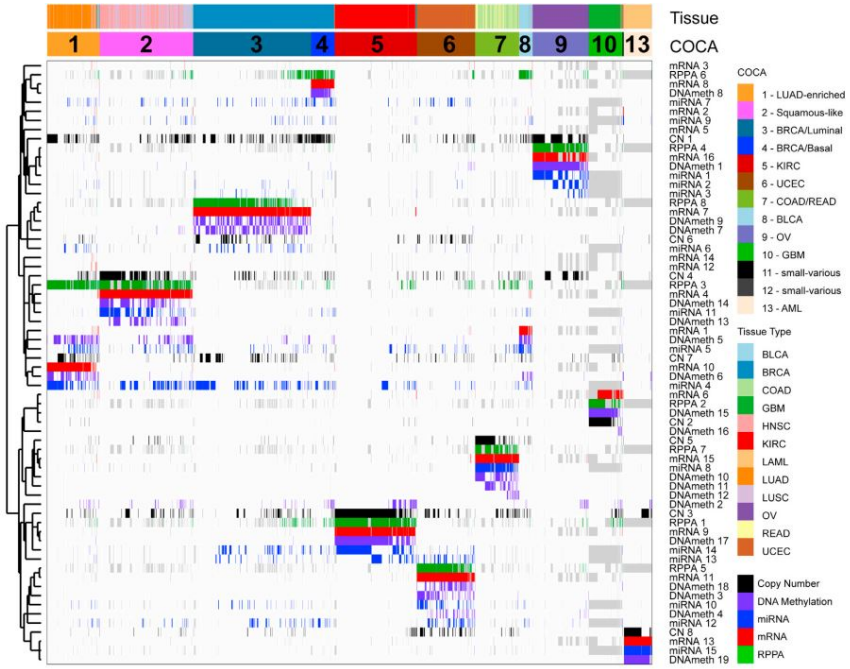
Cluster of Cluster Assignments, in short COCA, by Katherine A. Hoadley et al.[5] is a late integration method, clustering TCGA samples from 12 different cancer tissues made as part of The Cancer Genome Atlas Research Network.
It is composed of two steps. First, it clusters each omic data separately. Each omic can be clustered using different algorithms and hyperparameters, including k, the number of clusters. Second, for each sample's omic, cluster membership is collected in the form of an indicator vector. These vectors are then concatenated to represent the sample across all omics. It then runs consensus clustering on the samples' indicator representation, sampling 80% of the dataset and using a hierarchical clustering algorithm on the result.
For example, assume sample i is in cluster 3 out of 5 in omic 1 and in cluster 1 out of 3 in omic 2. The indicator vector for omic 1 is (0,0,1,0,0) and for omic 2 the vector is (1,0,0). Thus, the final representation of sample i is (0,0,1,0,0,1,0,0).
Some of the advantages in this representation is not needing to normalize the combined data and preventing omics with far more features to overshadow and dominate omics with less features.

Reminder: Consensus Clustering was discussed in further details when it was introduced. The algorithm samples and clusters the dataset on multiple iterations.
In each iteration a subset of the samples is taken and clustered. For each sample pair i and j, it records if sample i and j were sampled, denoted by $I^m(i,j)$, and if they were assigned to the same clustered, denoted by $M^m(i,j)$. At the end of the iterations we compute distance between each pair, $D(i,j)$, to be the frequency they weren't clustered together. Meaning, $D(i,j) = 1 - \sum_m M^m(i,j) / \sum_m I^m(i,j)$.
Finally we cluster of the samples based on $D(i,j)$, and return it as the consensus clustering.

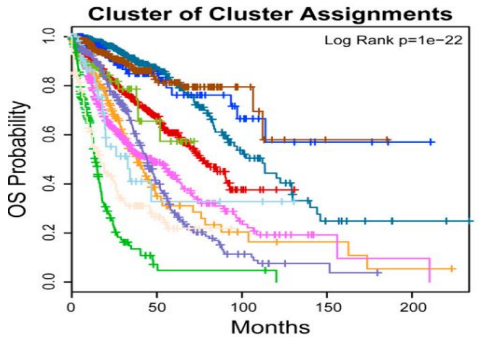
9.2.2 Results

COCA was used on 3527 TCGA samples from 12 tissues with 5 different types of omics: gene expression, methylation, miRNA, copy number variation and RPPA (protein arrays). Each omic was clustered using a different algorithm. It resulted in 13 clusters, 2 of them were excluded due to low sample count (< 10). 5 out of the 11 had nearly one-to-one relationships with the tissue of origin. To show the difference between clusters survival analysis was performed and suggested a difference between the clusters.
The main result of the method was that the clusters do not perfectly match the tissue of origin. First, lung squamous and head and neck were clustered together, which may reflect similar cell type of origin (head and neck cancers also appear in squamous cells) or explained

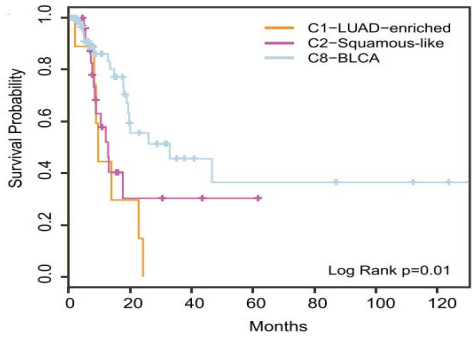
by smoking as an etiological factor; Second, bladder cancer was split across 3 pan-cancer subtypes, but survival analysis of bladder samples from different clusters suggests a difference between the tissue samples (log rank p=0.01).



(a)



(b)



(c)

Figure 9.2: **(a)** COCA results. The matrix is column indicator vectors of the samples. Each row is a cluster from an omic ordered by the consensus clustering hierarchy. Each cell is colored if the sample was grouped to that cluster. Grey coloring means missing data for that omic. The clusters are identified by the number and color in the second bar, and the top bar specifies the tissue of origin. **(b)** KM plot for all clusters, using the coloring of the top legend in (a). **(c)** KM plot of the bladder samples in the 3 clusters. Source:[5].

9.3 iCluster

9.3.1 Introduction

iCluster, by Ronglai Shen, Adam Olshen and Marc Ladanyi[6], is an early integration method based on the idea that tumor subtypes can be modeled as unobserved variables that can be simultaneously estimated from different omics. Using this idea, cancer data of n patients with p_i features from omic i, X_i, can be assumed to come from k subtypes of cancers. Thus, the data can be represented as a matrix factorization of cluster membership binary matrix which is shared across all omics and a coefficient matrix of that omic's features.
Formally, let X_i be a matrix of dimension $p_i \times n$. It can be represented as: $X_i = W_i Z + \epsilon_i$. Where Z is the cluster membership binary matrix of dimension $k \times n$, meaning its columns are standard basis vectors, $Z_{.j} = \big(0, \cdots, 0, 1, 0, \cdots, 0\big)^t \in \{0,1\}^k$. That means each sample can belong to only one cluster and since Z is shared across all omics, cluster membership is consistent in all omics. ϵ_i is a Gaussian noise added per column with zero mean and diagonal covariance. That is, each feature in X_i has different independent noise.

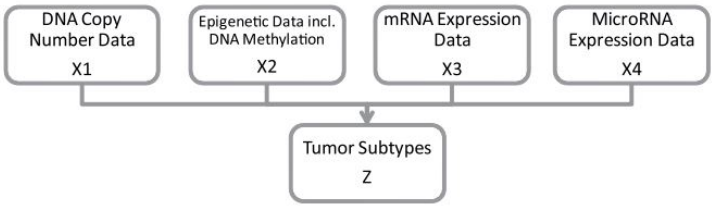


Figure 9.3: The integrative model. The tumor subtypes are represented by the joint variable Z that needs to be simultaneously estimated from multiple genomic data types measured on the same set of tumors. Source:[6].

9.3.2 Bayesian Statistics

In Bayesian statistics parameters are random variables, unlike frequentist statistics where parameters are numbers. For example, in frequentist statistics tossing a coin n times with probability p treats p as a number, while in Bayesian statistics p would come from some distribution.
Assume we want to choose the most probable p based on the data, that is maximize $P(p|data)$. In frequentist statistics, maximizing $P(p|data)$ makes no sense because p is not a random variable, thus we estimate p by maximizing the likelihood $P(data|p)$.
In Bayesian statistics, we can use Bayes' rule to maximize p, $P(p|data) = P(data|p) * \frac{P(p)}{P(data)} \underset{\text{P(data) is constant}}{\propto} P(data|p) * P(p)$.

9.3.3 iCluster

As mentioned in section 9.3.1, we define each omic data $X_i = W_i Z + \epsilon_i$. We assume ϵ_i has zero mean and a diagonal covariance matrix ψ_i.
An issue with our current model raises is that Z is discrete, making it hard to compute. Because of that we use a continuous representation Z^*, with multivariate normal prior distribution $Z^* \sim N(0, I)$.

Let m be the number of omics. Denote $X = (X_1, ..., X_m)^T$ and $W = (W_1, ..., W_m)^T$. It can be shown that X is a multivariate normal distribution:

$$X = (X_1, ..., X_m)^T \sim N(0, WW^T + \psi)$$

We now have our hidden variable Z^*, our data X and our model $\theta = (W, \psi)$. Using the log likelihood won't give us Z^* since it's not a variable there. Instead we write the complete log likelihood $l_c(X, W, \psi, Z)$ and try to optimize the problem using EM.

$$l_c(X, W, \psi, Z) = P(X, Z; \psi, W) \underset{Bayes'}{=} P(X|W, \psi, Z) * P(Z)$$

Z and X given W, ψ, are both multivariate normal distributions, so their density function given by the multivariable normal density function:

$$X \sim N(\mu, \Sigma) \rightarrow f_X(x) = det(2\pi\Sigma)^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)}.$$

Which makes our log likelihood to be:

$$l_c(X, W, \psi, Z) = -\frac{n}{2}\left[\sum_{i=1}^{m} p_i ln(2\pi) + ln(det(\Psi))\right] - \frac{1}{2}\left[tr((X - WZ^*)^T \psi^{-1}(X - WZ^*)) + tr(Z^{*^T} Z^*)\right]$$

Reminder: EM Expectation Maximization was discussed in regulatory motif discovery, specifically in MEME algorithm. EM starts with initial model θ and repeat two steps until θ converges:

- E-step - Re-estimate Z from θ, X.
- M-step - Re-estimate θ from X, Z

Note that the number of parameters for this optimization problem is $O(p) >> n$, which may cause overfitting, making a sparse solution to W more desirable. For that we add Lasso regularization[7] which penalizes the likelihood for any coefficient in W that is non-zero, thus encouraging the model to use less features. Lasso regularization has a hyperparameter $\lambda > 0$ which represent the trade off between nullifying features and maximizing the likelihood.

$$l_{c,p}(W, \psi, Z) = l_c(W, \psi, Z) - \lambda * \sum_{i=1}^{m} \sum_{k=1}^{K-1} \sum_{j=1}^{p_i} |w_{ikj}|$$

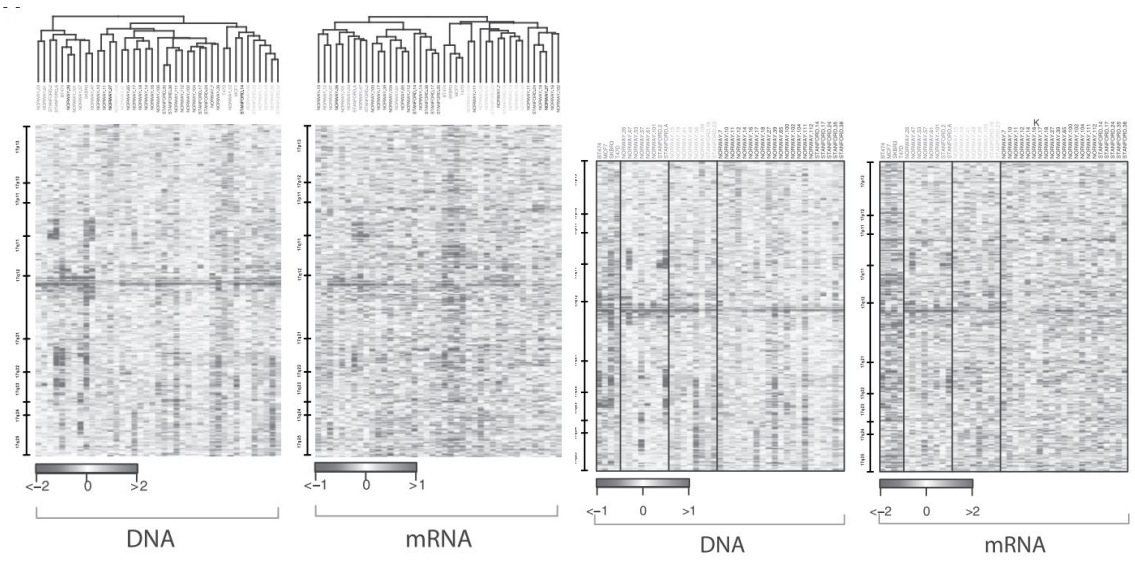
In the case of iCluster's model, the E-step provides a simultaneous dimension reduction by mapping the original data matrices of dimensions $(p_1, ..., p_m) \times n$ to a substantially reduced subspace represented by Z^* of dimension $K \times n$.

$$\begin{aligned}
\text{E-step:} \quad & E[Z^*|X] = W^T(WW^T + \psi)^{-1}X \text{ and} \\
& E[Z^*Z^{*^T}|X] = I - W^T(WW^T + \psi)^{-1}W + E[Z^*|X]E[Z^*|X]^T \\
\text{M-step:} \quad & \psi^{(t+1)} = \frac{1}{n} diag\{xX^T - W^{(t)}E[Z^*|X]X^T\} \text{ and} \\
& W_{lasso}^{(t+1)} = sign(W^{(t+1)})(|W^{(t+1)}| - \lambda) \\
\text{Where} \quad & W^{t+1} = (XE[Z^*|X]^T)(E[Z^*(Z^*)^T|X])^{-1}
\end{aligned}$$

Finally, we run k-means on $E[Z^*|X]$ to obtain Z. Each sample's cluster membership is based on the k-means result, giving us the columns of Z.
In order to determine the best λ and K, after we calculate Z^* and Z we measure the distance of absolute values between $E[Z^*|X]^T E[Z^*|X]$ which is the normalized observed values to $E[Z|X]^T E[Z|X]$, a perfect 1-0 block matrix, indicting whether two samples belong to the same cluster. λ, K are chosen so that the distance is minimal.

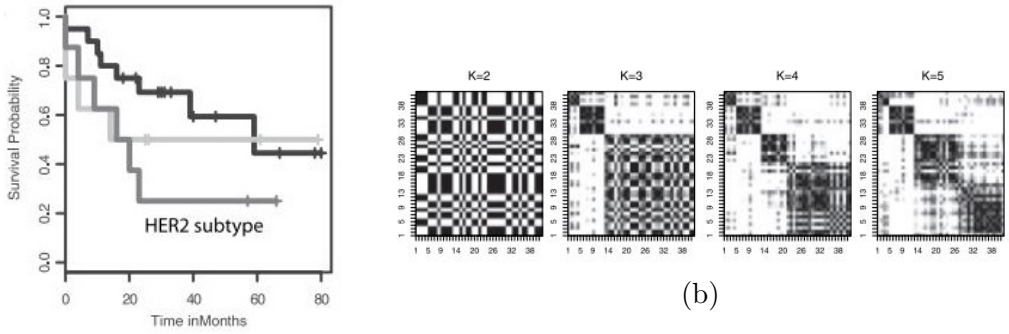
9.3.4 Results

The dataset contains 37 breast cancer patients and 4 cell lines samples (synthetic cancer tissue that can be grown in a lab. outside a human body). The omics used are: gene expression and copy number variation.
iCluster was tested against separate omic hierarchical clustering. After parameter optimization, iCluster found 4 distinct clusters, one for cell lines and three for the breast cancer patients, shown in Figure 9.4. The 4 cell lines were clustered together and separately from the rest of the samples, while in the hierarchical clustering of copy number variation they were scattered across the tree.
To show distinction between clusters, survival analysis between the 3 clusters of breast cancer patients was performed, showing different survival rate.



(a) (b)

Figure 9.4: Results in (a) and (b) are viewed by each omic. To the left is copy number variation, labeled DNA in both (a) and (b). To the right is gene expression, labeled mRNA in both (a) and (b). **(a)** The results from separate hierarchical clustering each omic. Cell line samples are scattered across the hierarchical clustering in copy number variation (left). **(b)** The results of iCluster, the small left cluster containing 4 samples are the cell lines samples. Source:[6].



(a) (b)

Figure 9.5: **(a)** KM plot of the 3 breast-cancer clusters in iCluster, showing distinct survival rate. **(b)** Cluster separability plots ($E[Z^*|X]^T E[Z^*|X]$). For k=4 and k=5 the matrix resembles a 1-0 blocks matrix. Source:[6].

9.4 Joint NMF

9.4.1 Introduction

Joint NMF, by Shihua Zhang, Jasmine Zhou et al.[8], is an algorithm based on the dimension reduction algorithm *Non-negative Matrix Factorization* (NMF)[9]. Joint NMF doesn't attempt to cluster the data but rather to find *multi-dimensional modules* (md-modules). Similar to co-modules in PING-PONG and biclusters in general, md-module is a subset of features from different omics that all or some of the samples exhibit correlated profiles across. The md-modules may overlap in features and samples.
Identifying modules can help break down massive sets of data into smaller ones that exhibit similar patterns, capturing associations between different omics while reducing the complexity of the data. It can also be used to differentiate between groups of patients.

9.4.2 Non-negative Matrix Factorization

NMF is a matrix decomposition problem of a non-negative matrix into a product of two non-negative matrices. One of the most common algorithms to solve this problem is Lee and Seung's multiplicative update rule[9].

Formally, Let matrix $X \in M^{n\times m}, X \geq 0$. We wish to find $W, H \geq 0$ s.t. $X = WH$.
The matrix multiplication can be implemented as computing the column vectors of X as linear combinations of the column vectors in W using coefficients supplied by rows of H. That is, each column in X can be computed as follows:

$$x_{\cdot j} = \sum_l w_{\cdot l} h_{lj} = W h_{\cdot j}$$

The error function used for NMF is:

$$min_{W,H}\|X - WH\|_F, \quad \|A\|_F = \sqrt{\sum_i \sum_j a_{ij}^2}$$

Lee and Seung's Algorithm:

- Initialize W, H non-negative matrices.
- In iteration n do:

  $$H_{ij}^{n+1} = H_{ij}^n \frac{((W^n)^T X)_{ij}}{((W^n)^T W^n H^n)_{ij}} \qquad W_{ij}^{n+1} = W_{ij}^n \frac{(X(H^{n+1})^T)_{ij}}{(WH^{n+1}(H^{n+1})^T)_{ij}}$$

  Where H^n, W^n are the current matrices in iteration n.
  Halt when W and H are stable, that is the error function is smaller than a predefined ϵ or n exceeds the maximum number of iterations.

9.4.3 Joint NMF

In our context, assume we want k md-modules. Each omic is represented by a matrix $X_l \in \mathbb{R}^{M \times N_l}$, with the M patients and N_l features.

Preprocessing Stage: We normalize X_l per feature, which leads to negative cells in the matrix. In order to fix that, we double the number of columns. For each feature, one column contains only the positive values and zero for negative values and the second column zeros the positive values and keep the absolute value of the negative cells. This makes our normalized matrix non-negative once again.

W is a $M \times k$ matrix, and its columns are used as basis vectors across all omics. H_l is the coefficient $k \times N_l$ matrix of omic l. Meaning, we have the same basis vectors W with k column vectors and each omic differs in the coefficient matrix. We need to change the error function accordingly to minimize distance for each omic:

$$min \sum_l \|X_l - W H_l\|_F^2$$

We also need to alter the update rule, updating W and each H_l:

$$(H_l^{n+1})_{ij} = (H_l^n)_{ij} \frac{((W^n)^T X_l)_{ij}}{((W^n)^T W^n H_l^n)_{ij}} \qquad W_{ij}^{n+1} = W_{ij}^n \frac{(\sum_l X_l (H_l^{n+1})^T)_{ij}}{(W \sum_l H_l^{n+1} (H_l^{n+1})^T)_{ij}}$$

Once W and H_l are optimized, in order to associate features with md-modules we normalize each H_l by row using z-score, $z_{ij} = \frac{H_{ij} - \mu_i}{\sigma_i}$. We include in the module only features with z-score that have exceeded a threshold. In the same manner, to associate patients with a module we normalize each column in W and include patients with z-score exceeding a threshold.
The output are these k md-modules, with the associated features and patients.

9.4.4 Results

Joint NMF was performed on 385 samples of ovarian cancer data from TCGA, using gene expression, methylation and miRNA expression. k=200 was selected. The 200 md-modules covered in total 2985 genes, 2008 methylation sites and 270 miRNA. Each md-module had an average of 239.6 genes, 162.3 methylation sites and 14 miRNA, indicating a high overlap between modules.
Analysis of the dimension reduction was shown to capture most of the information embedded in the original data. Average sample correlations of the reconstructed data using the md-modules and the original data per omic were about 0.91, with small variance, demonstrating robustness of the method.

Figure 9.6: Box-plot of sample-wise correlations of original and reconstructed omics Source:[8].

To assess the biological relevance of the md-modules, functionally homogeneous ratio of the members of individual omics was tested against random modules. A set of features was defined functionally homogeneous if it was enriched with at least one GO term. For each module, features from each omic were tested individually and all omics were tested combined. The combined omics showed higher ratio of enrichment than any omic individually, giving evidence to the importance of the multi-omic approach.
Furthermore, md-modules were tested for relevance in cancer study. 22 md-modules were found enriched with known cancer related genes, while the expected number by chance is 10. 20 md-modules contained patients with significantly different age characteristics than patients not in the module, and survival analysis showed a difference between patients in some modules compared to other patients. The ratio of survival difference between module and non-module was absent in the article.

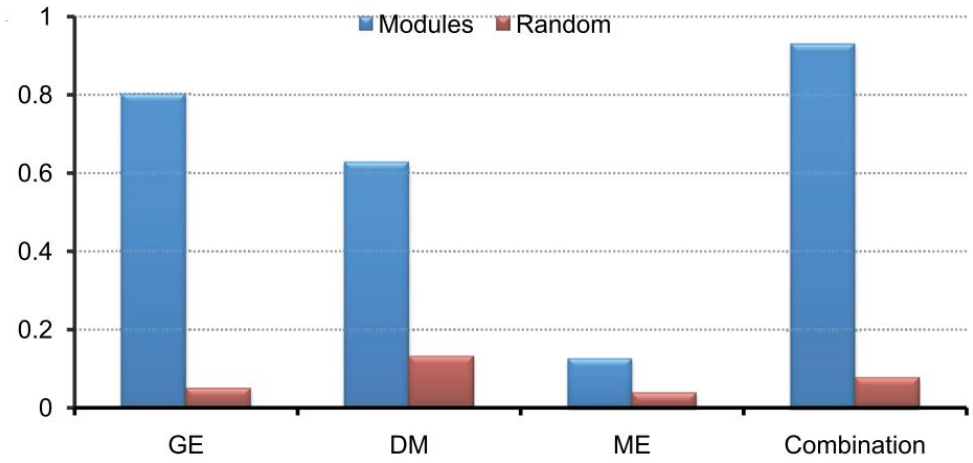


Figure 9.7: Enrichment ratio of md-modules in each omic, with respect to the GO terms, compared to the mean enrichment ratio of random runs. Source:[8].

9.5 Similarity Network Fusion

9.5.1 Introduction

Similarity Network Fusion, by Bo Wang et al.[10] (Anna Goldenberg's group), clusters patients based on a similarity network. Inspired by the theoretical multiview learning framework developed for computer vision and image processing applications[11], the algorithm constructs a similarity network for each omic and fuses them together. In a network, each node is a patient and the weight on an edge is the similarity between the patients in that omic. The algorithm then performs network fusion, by iteratively updating the weights, bringing the networks closer to each other until they are similar enough to converge into a final fused network.

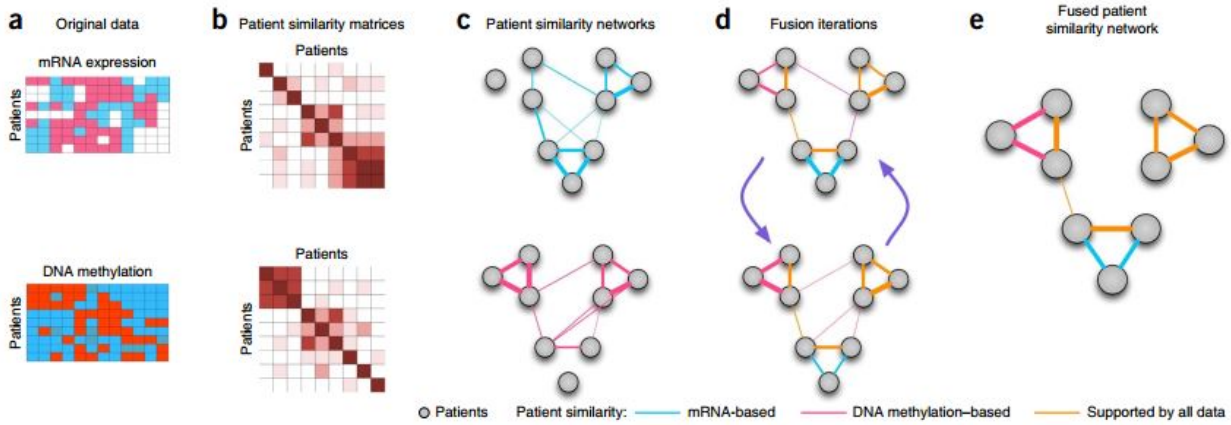


Figure 9.8: An example for the steps of SNF with two omics: methylation and mRNA expression. **(a)** We begin with a patients-to-features matrix per omic. **(b)** Compute the similarity matrix between patients for each omic. **(c)** Change the representation to a weighted graph. **(d)** Network fusion by iteratively updating the weights with information from the other networks, making them more similar with each step. **(e)** The iterative network fusion results in convergence to a single network. Source:[10].

Once the similarity between patients is computed, the algorithm is independent of the number of features originally in the omic and the integration's complexity depends solely on the number of patients. Since in genomics and cancer study the number of genes (features) is much higher than the number of patients, a similarity network has an advantage over other methods that attempt to address the problem with dimension reduction, because their complexity depends on the number of features and the methods are more sensitive to feature selection. On the other hand, features' roles aren't incorporated in the model and are therefore harder to interpret.

9.5.2 SNF algorithm

We first define the similarity matrix, W. Let $x_1, ..., x_n$ be a set of patients. SNF uses a scaled exponential similarity kernel, with Euclidean distance between patients, $\rho(x_i, x_j)$. That is,

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

where μ is a hyperparameter recommended in the range [0.3, 0.8]. In practice 0.5 was used. $\epsilon_{i,j}$ measures the average distance of x_i and x_j from their k nearest neighbours, denoted by N_i, N_j. It is used as a scaling parameter that controls the nodes' density.

$$\epsilon_{i,j} = \frac{mean(\rho(x_i, N_i)) + mean(\rho(x_j, N_j)) + \rho(x_i, x_j)}{3}$$

In order to compute the fused matrix from multiple types of measurements two similarity matrices are used. The first is P, a relative symmetric similarity matrix, where the sum of each row is 1 and the similarity of a node to itself is defined as 1/2. The second is S, a relative similarity within k nearest neighbours, meaning non-neighbouring points in S are set to zero. This is following an assumption that similarities to close neighbours are more reliable than to remote ones. Formally,

$$P(i,j) = \begin{cases} \frac{W(i,j)}{2\sum_{k\neq i} W(i,k)}, & j \neq i \\ 1/2, & j = i \end{cases} \qquad S(i,j) = \begin{cases} \frac{W(i,j)}{\sum_{k\in N_i} W(i,k)}, & j \in N_i \\ 0, & OW \end{cases}$$

Next, the algorithm iteratively fuses the networks, by updating P in each iteration. We only update P in each iteration, S remains static. Starting from P as the initial state using S as the kernel matrix in the fusion process, following the assumption that similarities to close neighbours are more reliable, this would capture the local structure of graphs, rather than the full structure. As a side note, it is also computationally more efficient.
Let m be the number omics. Let $P_t^{(v)}$ denote the relative similarity matrix P of omic v after t iterations. Then the updating rule for iteration t is:

$$P_{t+1}^{(v)} = S^{(v)} \times \frac{\sum_{k\neq v} P_t^{(k)}}{m-1} \times (S^{(v)})^T, \quad v \in \{1, ..., m\}$$

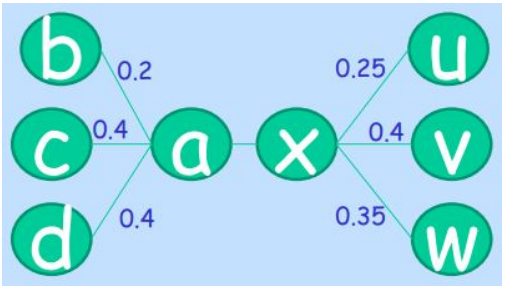
$$\rightarrow P_{t+1}^{(v)}(i,j) = \sum_{k\in N_i} \sum_{l\in N_j} S^{(v)}(i,k) * S^{(v)}(j,l) * \frac{\sum_{k\neq v} P_t^{(k)}(k,l)}{m-1}$$

At the end of each iteration, P_{t+1} becomes asymmetric, due to the asymmetry in S. We normalize P_{t+1} and make it symmetric again. After a fixed number of iteration, the fused network is set to:

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

Once we have a single fused network we use spectral clustering[12] to get the clusters from the similarity network.

For example, for m=2 at iteration t, given $S^{(1)}$ and a table of nearest neighbours of a and x in $P_t^{(2)}$, we would like to compute $P_{t+1}^{(1)}(a, x)$:



$S^{(1)}$

i	j	$P_t^{(2)}(i, j)$
b	u	0.02
b	v	0.007
b	w	0.01
c	u	0.09
c	v	0.08
c	w	0.003
d	u	0.05
d	v	0.008
d	w	0.03

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

$$\rightarrow P_{t+1}^{(1)}(a, x) = \sum_{k \in N_a} \sum_{l \in N_x} S^{(1)}(a, k) * S^{(1)}(x, l) * P_t^{(2)}(k, l)$$

$$= S^{(1)}(a, b) * S^{(1)}(x, u) * P_t^{(2)}(b, u) + S^{(1)}(a, b) * S^{(1)}(x, v) * P_t^{(2)}(b, v) + ...$$

$$= 0.2 * 0.25 * 0.02 + 0.2 * 0.4 * 0.007 + ...$$

9.5.3 Results

SNF was run on 3 types of omics: gene expression, methylation and miRNA, on 5 different types of cancer with 90-215 patients per type, from TCGA. k in SNF was chosen for each cancer type separately, and log-rank test was used to determine the significance of the results. On all cancer types SNF clustering using all omics showed a much higher statistical significance compared to SNF on each omic separately (Figure 9.9).
SNF was also compared to iCluster with different number of genes, using log-rank test, silhouette score to evaluate the coherence of the clusters and running time to evaluate scalability (Figure 9.10). SNF outperforms iCluster in all 3 categories.

Table 1 | SNF-based analysis versus individual data types

Cancer type	mRNA expression	DNA methylation	miRNA	SNF
GBM (3 clusters)	0.54	0.11	0.21	2.0×10^{-4}
BIC (5 clusters)	0.03	0.05	0.30	1.1×10^{-3}
KRCCC (3 clusters)	0.20	0.61	0.17	2.9×10^{-2}
LSCC (4 clusters)	0.06	0.26	0.46	2.0×10^{-2}
COAD (3 clusters)	0.18	0.04	0.46	8.8×10^{-4}

Analysis using Cox log-rank test *P* values.

Figure 9.9: Log-rank test of SNF on all omics compared to each individual omic. Source:[10].

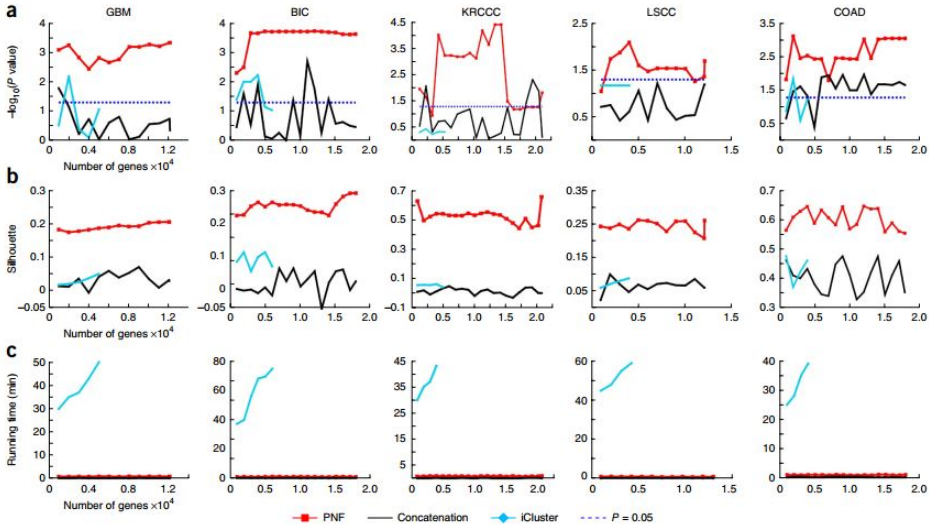


Figure 9.10: Comparison between SNF, iCluster and concatenation (early integration) in 3 categories across 5 cancer types as a function of the number of preselected genes (x axes). Genes were preselected based on significance in differential expression between tumor and healthy tissue in microarrays test.
(a) Cox log-rank test p value **(b)** Silhouette score **(c)** Run time comparison. Source:[10].

9.6 Multiple Kernel Learning

9.6.1 Introduction

Multiple Kernel Learning, by Nora Speicher and Nico Pfeifer[13], is a similarity based method that adapts the multiple kernel learning for dimensionality reduction framework[14](MKL-DR), which enables dimension reduction and data integration at the same time (using graph embedding[15] for dimension reduction). The general idea is to use several kernels on the omics. Each omic can use different kernels, which will give higher weight to the matrices with high amount of information while giving lower weight to those with low amount of information.
MKL-DR provides high flexibility with respect to the input data type, since the first step is to apply a kernel functions on the input data. Most importantly, multiple kernels can be used per data type.

9.6.2 Graph embedding

Graph embedding attempts to project the input vectors, $X = \{x_1, ..., x_N\}$, to a lower dimension while maintaining information about the similarities between vectors. Formally, we optimize based on the criterion:

$$\min_v \sum_{i,j=1}^{N} \|v^T x_i - v^T x_j\|^2 w_{ij} \qquad \text{subject to} \quad \sum_{i=1}^{N} \|v^T x_i\|^2 d_{ii}$$

Where W is the similarity matrix and D is a diagonal matrix representing constraints. We use D to avoid the trivial solution. Without it we can set v=0 to minimize the expression. Higher weight is given to vectors that are more similar to each other, therefore the algorithm will keep them closer together.

It can be shown that the optimal v is necessarily in the span of X. That is, $v = \sum_{n=1}^{N} \alpha_n x_n$. We can then use the kernel trick (reminder: $K(i,j) =< \phi(x_i), \phi(x_j) >$), thus,

$$v^T x_i - v^T x_j = \sum_{n=1}^{N} \alpha_n x_n x_i - \sum_{n=1}^{N} \alpha_n x_n x_j = \sum_{n=1}^{N} \alpha_n K(n,i) - \sum_{n=1}^{N} \alpha_n K(n,j)$$

9.6.3 Multiple Kernel Learning

It can be shown that a linear combination of kernels is also a kernel. We can set $K(n,i)$ to be $\sum_m \beta_m K_m(n,i), \quad \beta_m \geq 0$. $v^T x_i$ therefore equals:

$$\sum_{n=1}^{N} \alpha_n K(n,i) = \sum_{n=1}^{N} \alpha_n \sum_{m=1}^{M} \beta_m K_m(n,i) = \alpha^t K^i \beta$$

Where:

$$\alpha = [\alpha_1, ..., \alpha_N]^T \in \mathbb{R}^N$$
$$\beta = [\beta_1, ..., \beta_N]^T \in \mathbb{R}^M$$
$$K^i = \begin{pmatrix} K_1(1,i) & \cdots & K_M(1,i) \\ \vdots & \ddots & \vdots \\ K_1(N,i) & \cdots & K_M(N,i) \end{pmatrix}$$

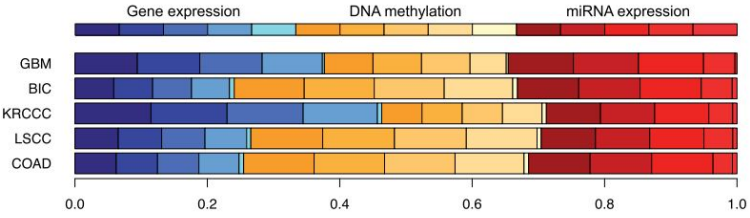
Which yields the following optimization problem:

$$\min_{\alpha,\beta} \sum_{i,j=1}^{N} \|\alpha^t K^i \beta - \alpha^t K^j \beta\|^2 w_{ij} \quad \text{subject to} \quad \sum_{i=1}^{N} \|\alpha^t K^i \beta\|^2 d_{ii}$$
$$\beta_m \geq 0. m \in \{1, .., M\}, \quad \|\beta\| = 1$$
$$w_{ij} = \begin{cases} 1, & i \in N_k(j) \vee j \in N_k(i) \\ 0, & OW \end{cases}$$
$$d_{ij} = \begin{cases} \sum_{n=1}^{N} w_{in}, & i = j \\ 0, & OW \end{cases}$$

We defined W and D with Locality Preserving Projections[16] method, which aims to conserve the distance of each sample to its k nearest neighbours, denoted by $N_k(i)$.
A constraint on β was added, $\|\beta\| = 1$, to avoid overfitting. It also helps us understand the proportional effect each kernel has.
As a last step, in order to cluster the data, we use k-means on the projection $\alpha^t K^i \beta$.

9.6.4 Results

The algorithm was tested against state of the art methods, for robustness and for clinical implications from the clustering results.
The state of the art method of choice was SNF, which is also similarity based. The same dataset from SNF's article is used[10]. That is, 3 types of omics are used: gene expression, methylation and miRNA on 5 different types of cancer with 90-215 patients each.
For each omic, the algorithm was run in two scenarios: either with one kernel per omic or 5 per omic, each with a Gaussian radial basis kernel function: $K(x,y) = exp(-\gamma\|x-y\|^2)$, $\gamma = \frac{1}{2d^2}, \gamma_n = c_n\gamma, c_n \in \{10^{-6}, 10^{-3}, 1, 10^3, 10^6\}$, where d is the number of features in the omic.
The number of dimensions for projection was fixed to 5 and the number of clusters for the k-means was chosen based on silhouette score.
As mentioned in subsection 9.6.3, the β values measure the effect of each kernel. Figure 9.11a shows the contribution of each kernel in each cancer type. Figure 9.11b shows a survival analysis comparison with SNF, showing a better performance than SNF and an increase in significance when using five kernels per omic instead of one.



(a)

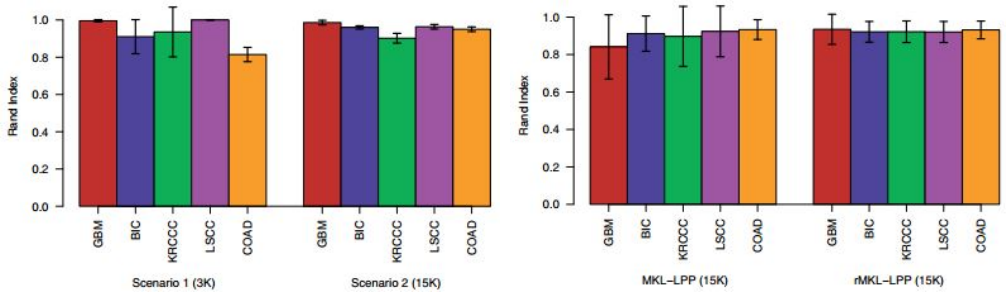
Cancer type	SNF	rMKL-LPP	
		3K	15K
GBM	2.0E-4 (3)	4.5E-2 (5)	6.5E-6 (6)
BIC	1.1E-3 (5)	3.0E-4 (6)	3.4E-3 (7)
KRCCC	2.9E-2 (3)	0.23 (6)	4.0E-5 (14)
LSCC	2.0E-2 (4)	2.2E-3 (2)	2.4E-4 (6)
COAD	8.8E-4 (3)	2.8E-2 (2)	2.8E-3 (6)

(b)

Figure 9.11: **(a)** Contribution of the different kernels (β values). **(b)** Survival analysis of clustering results of SNF and MKL with one and five kernels per data type. The numbers in brackets denote the number of clusters. Source:[13].

Assessing the robustness of the approach to small changes in the dataset, leave-one-out cross-validation was performed using Rand index[17] compared with the clustering of the whole dataset. In Figure 9.12a we see the stability of the clustering when using one kernel (Scenario 1) and five kernels (Scenario 2). The results seem more stable when using five kernels, which lowers the variance and in most cases increases the mean.
Testing the regulation constraint on β ($\|\beta\| = 1$), robustness was compared with and without the constraint. When using the constraint the results were more stable, showing lower variance and higher mean, giving evidence to the claim that without the constraint, the algorithm is overfitting.

To gain insights into the clinical implications of the identified clusters, survival analysis of the GBM cancer type patients (an aggressive brain cancer) was performed. Each cluster was split into two groups: those who were treated with Temozolomide, a chemotherapy drug for brain cancers; and those who weren't. In Figure 9.13 it can be seen that the treatment was effective only in some of the clustered groups, suggesting medical implications of the method.



(a) (b)

Figure 9.12: **(a)** Robustness of clustering for one and five kernels per omic with leave-one-out datasets measured using Rand index. **(b)** Robustness of clustering with and without constraint on β with leave-one-out cross-validation measured using Rand index. Source:[13].

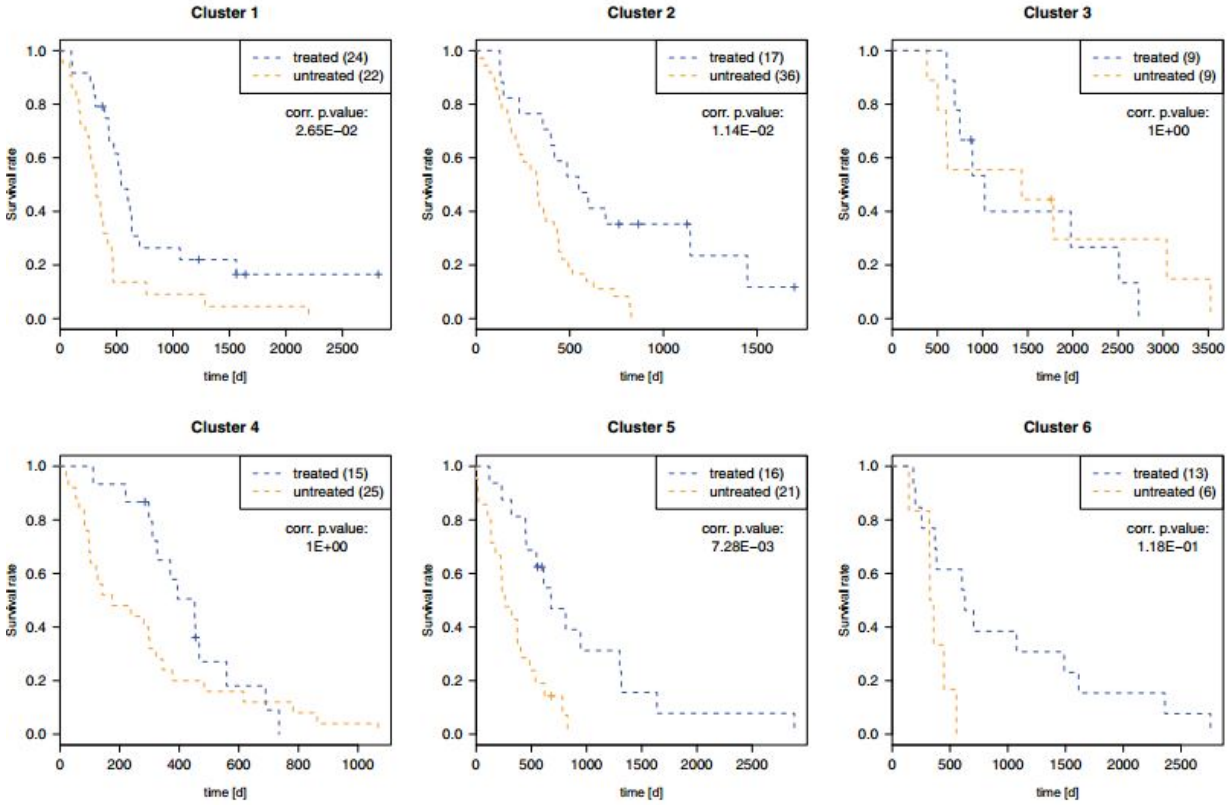


Figure 9.13: Survival analysis of GBM patients for treatment with and without Temozolomide in the different clusters. Source:[13].

Bibliography

[1] Steve Horvath. Dna methylation age of human tissues and cell types. *Genome Biology*, 14:3156, 2013.

[2] Scientific Creative Quaterly. `http://helicase.pbworks.com/w/page/17605615/DNA%20Methylation`, 2008.

[3] Francis Collins. Microrna research takes aim at cholesterol. `https://directorsblog.nih.gov/2013/11/26/microrna-research-takes-aim-at-cholesterol/`, 2013.

[4] What is a copy number variant, and why are they important risk factors for asd? `http://readingroom.mindspec.org/?page_id=8221`.

[5] Katherine A. Hoadley et al. Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell*, 158(4):929 – 944, 2014.

[6] 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. *Bioinformatics*, 25(22):2906–2912, 2009.

[7] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011.

[8] Shihua Zhang, Chun-Chi Liu, Wenyuan Li, Hui Shen, Peter W. Laird, and Xianghong Jasmine Zhou. Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Research*, 40(19):9379–9391, 2012.

[9] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562. MIT Press, 2000.

[10] Bo Wang, Aziz M. Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, 11:333 EP –, Jan 2014.

[11] B. Wang, J. Jiang, W. Wang, Z. H. Zhou, and Z. Tu. Unsupervised metric fusion by cross diffusion. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2997–3004, June 2012.

[12] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2002.

[13] Nora K. Speicher and Nico Pfeifer. Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics*, 31(12):i268–i275, 2015.

[14] Y. Y. Lin, T. L. Liu, and C. S. Fuh. Multiple kernel learning for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6):1147–1160, June 2011.

[15] S. Yan, D. Xu, B. Zhang, H. j. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, Jan 2007.

[16] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*, pages 153–160. MIT Press, 2004.

[17] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.