Costa Rica Institute of Technology

School of Computing
Master's Program in Computing

Clustering of Cases from Different Subtypes of Breast Cancer Using a Hopfield Network Built from Multi-omic Data

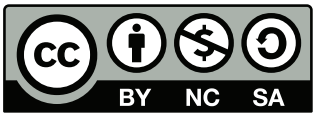
Thesis submitted to the School of Computing,
to opt for the degree of Magister Scientiae in Computing,
with emphasis in Computer Science

Olger Calderón Achío
Student

Dr. rer. nat. Francisco Siles Canales
Thesis Adviser

San José, November 2018

TEC | Tecnológico de Costa Rica



www.tec.ac.cr

TEC | Tecnológico de Costa Rica
Programa de Maestría en Computación
Teléfono: 2550-2402

SINAES POST-GRADO ACREDITADO COSTA RICA

APROBACIÓN DE PROYECTO

"Clustering of Cases from Different Subtypes of Breast Cancer Using a Hopfield Network Built from Multi-omic Data"

TRIBUNAL EXAMINADOR

Dr. Francisco Siles Canales
Profesor Asesor

Dr. Esteban Meneses Rojas
Profesor Lector

Dr. Rodrigo Mora Rodríguez
Profesor Externo

Dr. Roberto Cortés Morales
Coordinador del Programa
de Maestría en Computación

TEC | Tecnológico de Costa Rica
Maestría en Computación

Noviembre, 2018

Dedication

To my parents and brother, who remained by my side from beginning to end and whose support was always unconditional.

Acknowledgements

I would like to thank Dr. Francisco Siles, my thesis advisor and director of the Pattern Recognition and Intelligent Systems Laboratory, for his constant reminder of "persevering under pressure". This phrase has a personal and special meaning for both of us.

I am grateful to Dr. Esteban Meneses from the National Advanced Computing Collaboratory (CNCA) and Dr. Rodrigo Mora from the Tumoral Chemosensibility Laboratory (LQT), who showed interest in my research and made me feel like a colleague of theirs. Hope this collaboration might not be the last one.

A very special gratitude goes out to the members of the PRIS-Lab (specially those of BEND team) who made me feel part of a greater research initiative. Tackling complex problems like cancer does not sound that scary alongside them.

And finally, last but by no means least, to the colleagues at Mobilize who identified with me and supported my goals in one way or another. Special mention to Diego Pérez for being the best working partner at the master's program. My gratitude towards Carlos Bastos, Olman García, Ivan Sanabria and Hugo Fernández for their comprehension along the way.

Epigraph

The results suggest a helical structure.

Rosalind Franklin, *Lecture Notes*
(November 1951)

Abstract

Despite scientific advances, breast cancer still constitutes a worldwide major cause of death among women. Given the great heterogeneity between cases, distinct classification schemes have emerged. The intrinsic molecular subtype classification (luminal A, luminal B, HER2-enriched and basal-like) accounts for the molecular characteristics and prognosis of tumors, which provides valuable input for taking optimal treatment actions. Also, recent advancements in molecular biology have provided scientists with high quality and diversity of omic-like data, opening up the possibility of creating computational models for improving and validating current subtyping systems. On this study, a Hopfield Network model for breast cancer subtyping and characterization was created using data from The Cancer Genome Atlas repository. Novel aspects include the usage of the network as a clustering mechanism and the integrated use of several molecular types of data (gene mRNA expression, miRNA expression and copy number variation). The results showed clustering capabilities for the network, but even so, trying to derive a biological model from a Hopfield Network might be difficult given the mirror attractor phenomena (every cluster might end up with an opposite). As a methodological aspect, Hopfield was compared with kmeans and OPTICS clustering algorithms. The last one, surprisingly, hints at the possibility of creating a high precision model that differentiates between luminal, HER2-enriched and basal samples using only 10 genes. The normalization procedure of dividing gene expression values by their corresponding gene copy number appears to have contributed to the results. This opens up the possibility of exploring these kind of prediction models for implementing diagnostic tests at a lower cost.

Index Terms - Breast Cancer, Neural Networks, Machine Learning, Pattern Recognition, Biocomputing.

Resumen

A pesar de los avances científicos, el cáncer de mama todavía constituye una de las principales causas de muerte entre las mujeres en todo el mundo. Dada la gran heterogeneidad entre los casos, han surgido distintos esquemas de clasificación. La clasificación intrínseca de subtipos moleculares (luminal A, luminal B, HER2-enriquecido y tipo basal) toma en cuenta las características moleculares y el pronóstico de los tumores, lo que proporciona información valiosa para realizar acciones de tratamiento óptimas. Además, los recientes avances en biología molecular han proporcionado a los científicos alta calidad y diversidad de datos ómicos, lo que abre la posibilidad de crear modelos computacionales para mejorar y validar los sistemas de subtipos actuales. En este estudio, se creó un modelo de Red de Hopfield para el subtipo y la caracterización del cáncer de mama utilizando datos del repositorio "The Cancer Genome Atlas". Aspectos novedosos incluyen el uso de la red como un mecanismo de agrupación y el uso integrado de varios tipos de datos moleculares (expresión genénica de ARNm, expresión de miARN y variación del número de copias). Los resultados mostraron capacidades de agrupamiento para la red, pero aun así, tratar de derivar un modelo biológico de una Red de Hopfield podría ser difícil dado el fenómeno de los atractores espejo (cada grupo podría terminar con un opuesto). Como aspecto metodológico, se comparó a Hopfield con los algoritmos de agrupamiento kmeans y OPTICS. El último, sorprendentemente, sugiere la posibilidad de crear un modelo de alta precisión que diferencie entre muestras luminales, enriquecidas con HER2 y basales usando solo 10 genes. El procedimiento de normalización de dividir los valores de expresión génica entre su número de copia de gen correspondiente parece haber contribuido a los resultados. Esto abre la posibilidad de explorar este tipo de modelos de predicción para implementar pruebas de diagnóstico a un costo menor.

Palabras Clave - Cáncer de Mama, Redes Neuronales, Aprendizaje de Máquinas, Reconocimiento de Patrones, Biocomputación.

Contents

Chapter 1

Introduction

1.1 Overview

Cancer is a complex disease of molecular basis with a high mortality rate. Given the great heterogeneity between cancer cases in terms of genetic behavior and prognosis, some authors prefer to think of "cancer" as a group of related diseases instead of a single pathological condition [29, 6, 16]. Such molecular diversity is even present on tumors originating from the same tissue. For example, breast cancer it is now known to have different molecular subtypes which the community has agreed upon, namely luminal A, luminal B, HER2-enriched and basal-like [44, 7]. Current classification schemes will rarely stay static as new studies might reveal finer-grained classifications for cancer subtypes. Better characterization of these subtypes at the molecular level will lead to the improvement of diagnostics and treatments.

Recent advancements in molecular biology have provided scientists with high quality and high diversity of omic-like data product of processes like sequencing, transcriptome profiling, copy number quantification and methylation profiling. This has opened the door to new types of studies which make use of computational models built upon this data. Such models look to capture patterns in data so our understanding in cancer dynamics can be improved. Sub-cancer classification has benefited from these approaches, specially from unsupervised learning techniques like clustering which has revealed the existence of novel cancer subtypes [43]. There is still room left for trying novel approaches in this regard, specially when making use of non-conventional clustering algorithms and of data integration techniques for different molecular data types, which overall, have showed promising results [38, 43].

An unsupervised learning neural network named Hopfield Network has just recently begun to be explored to model energy landscapes of cellular differentiation [41, 22, 13, 34, 12]. After being trained with samples data, this model generates a landscape with attractor states which

likely correspond to stable phenotypic states of cellular development, in a manner similar as Waddington Epigenetic Landscape does [13]. The states could describe either specialized healthy cells configurations (e.g: epithelial cells, neurons, blood cells), or diseases states as the different cancer subtypes [22, 13]. Although already been used to explore clustering of gene expression profiles into cancer subtypes [22], so far, the technique has not yielded a high quality characterization of breast cancer according to its molecular subtypes (luminal A, luminal B, HER2-enriched and basal-like). Also the training of these models has been strictly limited to using gene expression quantification data from microarrays.

On this study, it was explored if it was possible to create a Hopfield Network model for breast cancer subtyping and characterization. Several types of omic data from samples of The Cancer Genome Atlas were taken into disposition so different data integration strategies could be tested. While the initial expectation of finding finer subtypes aside from those exposed in the literature was rapidly turned down, the model showed capabilities for being used for clustering purposes. Still, some of its appealing features like the attractors landscape model should be accompanied of careful interpretation, as it might expose characteristics that make little sense under a biological context.

As a methodological aspect, Hopfield was compared against other clustering methods like kmeans and OPTICS (a density-based algorithm similar to dbscan). The last one, in a kind of unexpected way, showed the possibility of creating a high precision model that differentiates between luminal, HER2-enriched and basal samples using only a few genes. This opens up the possibility of exploring these kind of models for implementing diagnostic tests at a lower cost.

The present thesis document is composed of the following chapters:

- Chapter 1 serves as an introduction to the research problem at hand. Also, the contributions and objectives of this work are stated.

- Chapter 2 contains a theoretical framework with concepts relevant to both the research problem and applied methods.

- Chapter 3 describes most of the methodological aspects that were followed in order to fulfill the objectives.

- Chapter 4 shows and discusses the results obtained through experimentation.

- Chapter 5 enumerates some general conclusions, while providing ideas for future work in the same line of research.

1.2 Background and Related Work

1.2.1 Cancer Statistics

According to a study published by the American Medical Association, in the year 2015, it was estimated that 17.5 million cancer cases were diagnosed globally, and there were 8.7 million reported deaths [14]. It was the leading second cause of death behind cardiovascular diseases. Between 2005 and 2015, incidence of cancer increased by 33% due to factors like population growth and aging. These statistics reflect the need to invest in cancer control planning and prevention. This also calls for the development of accurate diagnostics, so patients can be treated in the most effective way for the particular cancer type/subtype they have.

In 2015, it is estimated that 2.4 million cases of breast cancer occurred and it was the cause of death for 523.000 women and 10.000 men [14]. In the context of Costa Rica and according to the Ministry of Health, in year 2010, breast cancer occupied the first place on mortality by cancer on women, followed by stomach, colon and cervix cancer [8].

1.2.2 Repositories of Molecular Data

In the latest years there have been great advancements in the bioinformatics community in terms of production of molecular data and available repositories.

The Gene Expression Omnibus is a public functional genomics data repository supporting data submissions, mainly gene expression profiles [11]. The Cancer Cell Line Encyclopedia (CCLE) provides detailed genetic characterization of human cancer cell lines including access to DNA copy number, gene expression and mutation data [3]. It also includes chemosensitivity responses for several anticancer drugs. The GenBank, the National Institute of Health (NIH) database of DNA sequences, is constantly being updated with annotated samples and makes a release every two months [4]. These public repositories bring new opportunities, encouraging the use of shared resources and scientific reproducibility.

The Cancer Genome Atlas (TCGA) is a project from the National Institute of Health that began in 2005 as an effort to recognize the genetic mutations that cause cancer. The aim was to open the door to new studies that could better describe the disease on its molecular terms [5]. So far, a lot patients have been part of the project (around 11.000 involving 33 different tumor types). The total data in TCGA is around 2.5 petabytes. For each patient it includes a variety of data sets product of processes like gene expression profiling, copy number variation profiling, SNP genotyping, genome wide DNA methylation profiling, microRNA profiling, and exon sequencing. The characterization of each case is comprehensive because of the possibility of using different molecular data types.

Just recently, the TCGA was integrated into a bigger initiative named Genomic Data Commons (GDC) [28]. The GDC mainly integrates two repositories: the TCGA and the Therapeutically Applicable Research to Generate Effective Therapies database (TARGET)[1]. Both databases have been harmonized under the same set of bioinformatics tools, opening the possibility of establishing comparisons between the two repositories of data. It is expected that new data will be imported to the GDC.

1.2.3 Clustering Gene Expression Patterns

The clustering of gene expression patterns from microarrays has been of special interest in the community. It opens up the door for gaining insights on gene co-regulation, and disease characterization as well [19]. New cancer types have been discovered and studied on this way.

Pioneering studies on this problem include [15] and [1]. On the former it was possible to discover the subtypes of acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) without prior knowledge using self-organizing maps (SOMs). In the second one, distinct types of diffuse large B-cell lymphoma were identified.

Average-linkage hierarchical clustering was used in [39], on gene expression patterns of breast cancer derived from cDNA microarrays. It was found that at least two distinct groups inside luminal type of cancer existed with characteristic gene expression profiles and different prognosis. The classification of luminal into A and B subtypes appears to initially come from this study.

There have been also comparative studies of the efficiency of several clustering algorithms under the context of gene expression clustering.

On [19], authors make note of the complexities associated with gene expression datasets like the high dimensionality and noise problems. The authors argue that agglomerative and partitive clustering techniques tend to fall short because of these aspects, and remark bi-clustering algorithms as an appealing option. An important discussion point is that clustering methods that don't need to specify an initial K (number of desired clusters) have an advantage in terms of exploration without *a priori* knowledge. But even if they don't need to specify K, some of them calculate thresholds for dividing the clusters which might make little sense in the biological context.

Souto et al performed a comparative study of seven clustering methods and four proximity measures for the analysis of 35 cancer gene expression microarray datasets [9]. The authors contributed with a large-scale evaluation which included other clustering methods aside from the "classic" ones like hierarchical clustering. The evaluated methods included: hierarchi-

[1]It puts emphasis on cases of childhood cancer.

cal clustering with single, complete, and average linkage, k-means, mixture of multivariate Gaussians, spectral clustering and nearest neighbor-based method. It was found that the finite mixture of Gaussians and k-means gave the best results in terms of revealing the true structure of the datasets. The authors also note that most of bioinformatics studies involving clustering make use of hierarchical clustering, probably because of factors like ease of use and available implementations.

Pirooznia et al compared the efficiency of several classification and clustering algorithms [32] making use of eight microarray cancer datasets. The evaluated clustering algorithms were: k-means, density-based clustering, farthest first traversal algorithm and expectation maximization clustering. Authors reported that the choice of feature selection method, the number of genes in the gene list, the number of cases (samples) and the noise in the dataset all influence the quality of the results. Farthest first traversal algorithm obtained the greatest accuracy.

Hopfield Networks just recently began to be used for clustering cancer samples into cancer subtypes by Maetschke et al [22].

Overall, most studies have been carried out with microarray data (vs RNA-Seq), non-neural networks algorithms and the number of used samples rarely overpass 100.

1.2.4 Hopfield Networks Usage in Gene Expression Landscapes

Hopfield Networks (HNs), recurrent neural networks that model associative memory [17, 23], have been used before to model attractor landscapes from gene expression data. The main idea has been to construct models that have the capacity of mirroring the dynamics of the underlying gene regulatory network when converging towards a certain stable state from the cell. Computing units in the network correspond to genes, and the weights between units represent how much the expression of genes affect each other.

The gene expression levels from a cell tend to converge to stable states of minimum energy as the cell differentiates into more specialized states [49] (see figure 1.1). In the same way, a HN can associate an input pattern with the most resembling stored pattern, hence converging in some way.

In [13], HNs are used to model the epigenetic landscape of cellular development for several biological processes in a manner similar to Conrad Waddington's model (which traces the evolution of cells from a pluripotent state to a committed state). The authors used gene expression data across different states of development in time. Using HNs, they identified genes and transcription factors that drive cell-fate transitions. A similar approach has been explored, but putting emphasis on modeling the progression of several diseases like Parkinson's disease, glioma and colorectal cancer [12]. Each attractor in the HN likely corresponds to a specific stage in the disease.

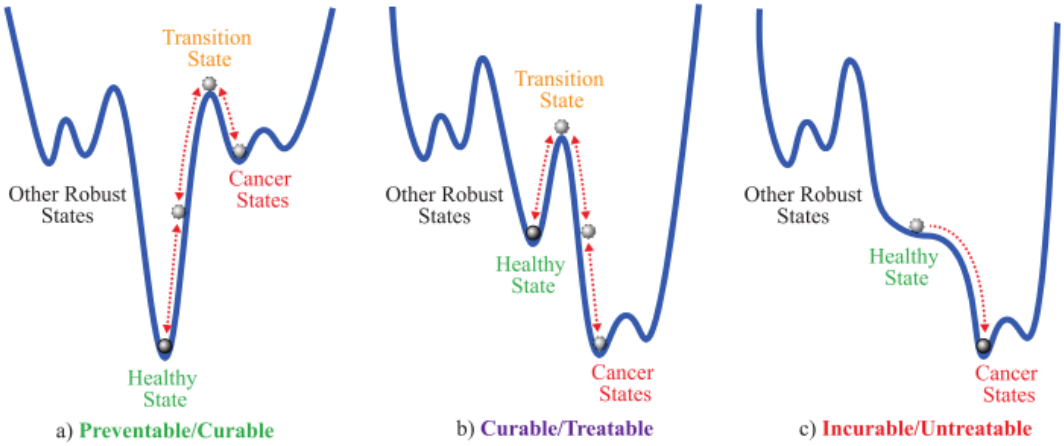


Figure 1.1: After being trained, a Hopfield Network can be used to model an energy landscape of cellular differentiation. Attractor states (stored patterns) at the bottom of the landscape correspond to stable phenotypic cell configurations of minimum energy (gene expression levels that induce the cell to either healthy or disease states). Transition states could be also present at peaks or as local minima. Source of image: [49].

Szedlak et al explored the asymmetric Hopfield model for generating attractor states for cancer gene expression data [41]. Once trained, the authors proposed an analysis on the network's weighted graph for detecting strongly connected cluster of nodes that have a significant impact on signaling in the gene network. In this way, candidate sets of proteins can be detected for intelligent therapeutic interventions.

Koulakov and Lazebnik used Hopfield Networks to support a model for cell fusion [21]. Each cell has a Hopfield Network with several attractors, each one corresponding to a stable cell phenotype. The fusion of two different cells might give rise to spurious attractor states which might turn the cell cancerous.

Pusuluri et al also study [34, 33] how recurrent neural networks can be used to model energy landscapes of biological processes with a given set of attractors. He puts particular attention to the properties of the generated landscapes instead of concentrating on other previously analyzed aspects of the networks like storage capacity and stability of patterns. Hopfield networks, and a similar model (Kanter and Sompolinsky) were used in the study.

1.2.5 Clustering Cases from Subtypes of Breast Cancer

The study from Sorlie et al [39] proves to be seminal in the area of breast cancer characterization using clustering algorithms. Using hierarchical clustering, they managed to detect several cancer subtypes inside the luminal one. In [46], a similar clustering approach was effectuated which resulted into two well defined clusters, which members differed in ER states and lymphocytic infiltration. Sotiriou et al also used hierarchical clustering in a

population-based study, giving as a result two major subgroups based on their ER status [40]. It correlated well with basal and luminal characteristics.

A comprehensive study using different data platforms from The Cancer Genome Atlas reinforces the fact that the four breast molecular subtypes (luminal A, luminal B, HER2-enriched and basal-like) are well established [44] and each possesses their own characteristics for each of the molecular subtypes. The authors claims that this reinforces the hypothesis that much of the clinically observable heterogeneity occurs within these major subtypes.

In [43] known breast cancer subgroups were detected. The authors made use of an integrated data approach of four platforms (gene messenger RNA expression, DNA-methylation, copy-number variation and microRNA expression) and clustered cases of 4,434 samples from The Cancer Genome Atlas across 19 cancer types. The technique involved the selection of principal components from each data type and then an early (concatenation-based) integration of data was performed. Then DBSCAN (Density-Based Algorithm for Discovering Clusters) was applied. Subtype analysis of the 563 breast cancer cases revealed five clusters that mostly correspond to the know molecular subtypes. However luminal A was splitted into two different clusters and some cases of basal-type were put together with the ones of HER2-type into a single cluster. See figure 1.2. This thesis work took a different approach for data integration, although it used the same molecular data types (except methylation values).

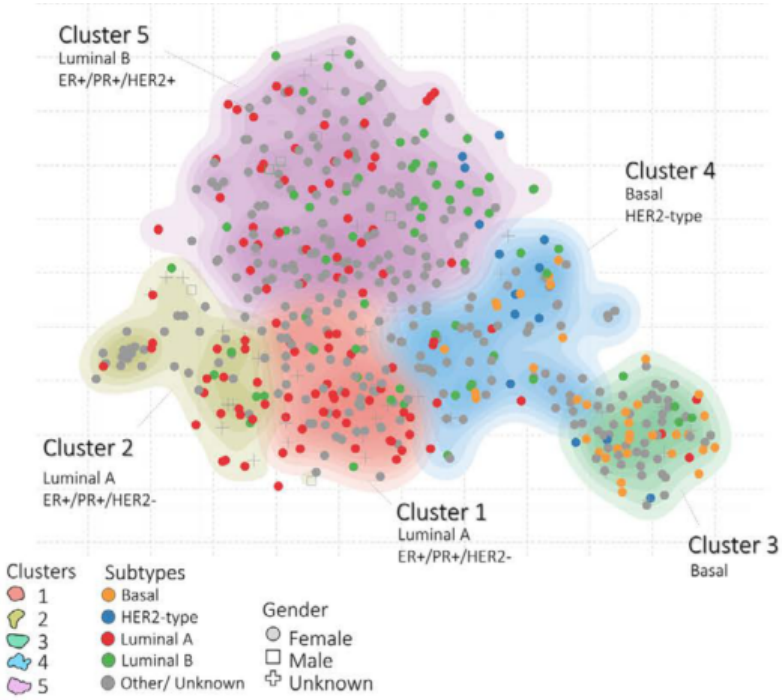


Figure 1.2: Clustering results from [43]. The method managed to divide luminal A cases into two clusters, but was unable to correctly separate HER2-type cases from basal.

The work from Maetschke et al [22] is the one that most resembles this one in terms of

objectives and methodology. They suggested using Hopfield Networks specifically for characterizing cancer subtypes as attractor states in a Hopfield Network. They also introduced a pruning method for discarding genes which would contribute little to the overall performance of the model. The authors used the microarray datasets proposed by Wang et al [47] and Souto et al [9] for training the networks. The ones that were trained using breast cancer datasets did not particularly show good results. This work differs in the sense that was specifically aimed at breast cancer characterization and contemplated using several omic types (not just gene mRNA expression) for improving the accuracy of the model. Used data was from the TCGA and, for validation purposes, prelabeled using the four main molecular subtypes of breast cancer.

1.3 Problem Definition

From the biological point of view, breast cancer heterogeneity constitutes the main problem. Aspects like diagnostics, prognosis and effectiveness of treatments are all affected by the molecular characteristics of tumors. This problem can be approached by building a computational model of breast cancer subtyping.

Given a dataset D with multi-omic data of n cancer samples, the problem of building a model for cancer subtyping can be formally defined as a type of clustering problem.

A clustering C for D is a partition of the n patients into mutually disjoint subsets C_1, C_2, ..., C_K called clusters. Formally the following properties must hold [25]:

$$C = \{C_1, C_2, \ldots, C_K\}, C_{k_1} \cap C_{k_2} = \emptyset \text{ with } k_1, k_2 \in \{1, \ldots, K\}, k_1 \neq k_2 \text{ and } \bigcup_{k=1}^{K} C_k = D$$

Also $|C_k| > 0$, $k \in \{1, \ldots, K\}$. In other words, empty clusters are not allowed.

There is a function *subtype* : $D \rightarrow L$ which associates a patient with a label for his/her cancer subtype. $L = \{l_1, l_2, \ldots, l_m\}$ is the set of possible labels known *a priori* for each known cancer subtype. In the context of breast cancer this set corresponds to:

$$L = \{luminalA, luminalB, basal, her2\}$$

.

These labels are know *a priori* for each patient.

A reference clustering C' can be defined as $C' = \{C'_1, C'_2, \ldots, C'_m\}$. Each cluster C'_k is defined as $C'_k = \{p_i \in D, i \in \{1, \ldots, n\} | subtype(p_i) = l_k\}$.

There is also a numerical function f that quantifies how much two clusterings differ according to some criteria. The problem consists in finding a clustering C such that $f(C, C')$ is optimized as much as possible. The function f chosen for this study was the Adjusted Rand Index (ARI) which ranges from -1 to 1. Higher values denote better agreement between the clusterings in terms of precision and recall.

From the statistical point of view, the problem can be seen as a multivariate analysis where thousands of variables (in this case transcript levels of genes) are involved per observation, and there is also the possibility of using other data views like copy number variation for deriving new features.

1.4 Hypothesis

Through the use of multi-omic data from the TCGA and a hebbian learning scheme, is possible to create a Hopfield Network model for breast cancer subtyping and characterization. This model will either differentiate between the main four molecular subtypes of breast cancer, or will find a finer classification inside these.

1.5 Justification

Under this context, having a computational model that accurately clusters patient cases would help in the characterization and discovery of subtypes of breast cancer. Hopfield Networks serve well both purposes. After being trained, each attractor state in the network encodes a binary state for each gene of interest (+1 for an highly expressed gene, -1 otherwise). An attractor defines a cluster for those cases that are converging to it.

Characterization means that key regulator genes and the dynamics between them could be better understood for each detected subtype (cluster) in the model. In other words, the attractor states of the model could aid in the identification of candidate biomarkers for prognosis and treatment, as such states would encode the different breast cancer configurations for the gene regulatory network. The candidate biomarkers would need to be validated later *in vitro* by experts in the area.

The model has also the potential to reveal new cancer subtypes. The unsupervised nature of the Hopfield Network model does not require prior knowledge, nor an initial number of clusters (k), and the attractor states are discovered as the network is trained. A set of

attractor states where a certain subtype of breast cancer cases are concentrated (e.g basal-like cases), might reveal a finer subdivision inside the subtype. This could be later validated using tools like survival analysis, which might reveal significant differences for the prognosis of the subtypes.

If compared to other classical approaches for clustering, there are a number of reasons why using Hopfield Networks represents an innovative and appealing option:

- **Biological interpretation**: Hopfield Networks associative memory capabilities can be used to model an energy landscape with attractor states of minimum energy at the bottom. The landscape metaphor has been used before by the biological community, specially when referring to the cellular differentiation process and the progression of diseases [49]. The cell's gene regulatory network, when adjusting the levels of transcripts of genes, highly resembles a dynamical system that seeks to converge to a stable state as time passes. Hopfield Networks, when recalling a stored pattern mimic exactly this behavior, as the input pattern converges to the state of the nearest attractor in the landscape.
- **No need of prior knowledge**: Hopfield Networks don't need an initial k number of clusters. Attractors are discovered based on the characteristics of the training data. This proves to be useful when looking for potential new cancer subtypes.
- **An unified clustering-classification framework**: Hopfield Networks separate the learning process (adjusting the weights of links between units) from the recalling process (associating an input with a stored attractor). The recalling process is first effectuated on the training samples for clustering purposes. Then when the generated model is validated and understood, recalling could be used on new samples for classification purposes.
- **Algorithmically simple**: The coding effort for implementing a Hopfield Network is not high. Between 35 and 50 lines of code are needed in a language like R which is optimized for data manipulation.
- **Other analysis**: It is possible to use visualization methods like the one in [22] for displaying the energy landscape associated with the network. This intuitively gives a measure of how far are cases from converging to their associated attractors. It is also possible to trace the convergence route from each case to their attractor, hence revealing the evolution of the gene regulatory network as it converges to a stable state [12]. Other properties of the attractors like their basin of attraction and density have been studied [33].

This study introduced some novel aspects from the methodological standpoint:

- The use of a Hopfield Network for clustering breast cancer cases into its four main molecular classes: luminal A, luminal B, basal-like and HER2-enriched. Other studies

involving Hopfield Networks are not fully focused on breast cancer and use other simpler classification schemes for validation.

- Testing new multi-omic approaches for data integration using three different views of data: gene (mRNA) expression, miRNA expression and copy number variation. Most studies are limited to the usage of gene (mRNA) expression only.
- A high quantity of cases from The Cancer Genome Atlas was used (around 1000). Most studies limit to smaller datasets (100 cases at most).

Other factors that also motivated and contributed to this research effort are the proliferation of data science techniques, the improvement of latest generation genomic and sequencing technologies and the collaborative effort that is being held by the PRIS-Lab[2] and the LQT[3], which played a key role for this project's definition and development.

The PRIS-Lab BEND team, in collaboration with other laboratories or research centers, is fully devoted to research on the field of Computational Biology. Research interests include: cell tracking on light field microscopy, chemosensitivity prediction, computational structural molecular biology, recognition of patterns in optical spectroscopy, among others. According to the National Institute of Health, the field of Computational Biology seeks to use mathematical and computational methods to answer theoretical and experimental questions in the sciences of life, in contrast with Bioinformatics which is more concerned about using informatics principles to make biological data more understandable and useful [30].

1.6 Objectives

1.6.1 Main Objective

Use a Hopfield Network model built from multi-omic data of the TCGA for clustering cases from different subtypes of breast cancer.

1.6.2 Specific Objectives

1. Create the necessary software infrastructure for loading and preprocessing the data from the TCGA.
2. Create the necessary software infrastructure for training and visualizing a Hopfield model.

[2]Pattern Recognition and Intelligent Systems Laboratory, School of Electrical Engineering, University of Costa Rica.

[3]Tumoral Chemosensibility Laboratory, Microbiology Department, University of Costa Rica.

3. Design and implement procedures for generating either standalone[4] or integrated datasets that make use of the gene (mRNA) expression, miRNA expression and copy number variation data views.

4. Run experiments varying the generated datasets, measure the effectiveness of the trained models (while comparing with at least two other algorithms) and analyze the results.

1.6.3 Scope

- The used data integration approaches were "early". This means new datasets were derived before performing the training procedure.

- Feature selection played a crucial role as the total number of features exceeded the 60000. Only protein coding genes were used and not more than 100 were chosen for creating a model.

- Even if the proposed approach for training the Hopfield model was general enough to be applied to data of other cancer types, the study was exclusively focused on breast cancer cases from the TCGA.

- Cases from the TCGA that did not have all the necessary data views (gene mRNA expression, miRNA expression, copy number variation) were excluded from the study.

- Having an initial subcancer PAM50 label for the used cases is a prerequisite condition for validating the created model. Otherwise, cases which did not possess this information were excluded from the study.

- It was not contemplated in the scope of the study creating a R package or similar.

- All analysis were executed *in silico*.

1.6.4 Deliverables

The following table describes all the deliverables for this project. Please refer to the specific objectives enumerated in the previous section.

[4]Standalone refers on this case to a dataset which contains information from a single type of data view.

Objective	Activities	Deliverable(s)
Specific Objective 1	Implement functions in R.	Source code in R for loading and preprocessing the data from the TCGA.
Specific Objective 2	Implement functions in R.	Source code in R for training and visualizing a Hopfield model.
Specific Objective 3	Define and implement procedures for generating either standalone or integrated datasets.	Source code in R for creating the datasets.
Specific Objective 4	Run experiments and collect results data.	Results from experiments and written analysis.

Chapter 2

Theoretical Framework

2.1 The Problem of Cancer

Cancer has been a mainstream human race health problem for years. It mainly causes an erratic behavior on the cell natural mechanisms of division and programmed death (apoptosis), which in turn triggers abnormal cell growth and proliferation through the tissues [29]. This process gives rise to the formation and development of masses of cells called tumors. It might prove lethal, specially if these enter an advanced state where they spread towards other organs different from the one of origin, a process called metastasis.

Carcinogenesis, the process of cancer formation, is mainly the result of mutations in somatic cells[1]. Some cancer-causing mutations are due to environmental factors and exposure to carcinogenic substances. Others are the result of errors during DNA replication and lack of proper DNA repairing processes. Given its genetic nature, the susceptibility to develop cancer varies from person to person. Cells can also become cancerous by being infected by certain viruses (called oncoviruses).

When a cell becomes cancerous, the gene expression levels of certain genes become altered and differ from that of a normal (differentiated) cell. There are mainly 2 types of genes that affect cancer. Oncogenes and tumor-suppressor genes [18, 16]. Oncogenes are mutated forms of certain genes that usually play some role in cell division or growth. By themselves, these genes (called proto-oncogenes in their normal forms), are necessary for the correct functioning of the cell. However, in their oncogene forms, they are cancer-producing agents. This is normally reflected in terms of proteins with hyperactive behaviors or overproduction of proteins associated with the oncogenes. Proto-oncogenes usually code for: growth factors, cell surface receptors, transcription factors and signal transmission proteins (like kinases).

[1] Cancer could start by other means like the silencing of key genes product of epigenetic modifications like methylation [42].

On other hand, tumor-suppressor genes usually suppress uncontrolled cell division. They have a protective function, that when inactivated via mutations, might give rise to tumors. It is usually necessary the combination of mutations on both oncogenes and tumor-suppressor genes to give rise to cancer. Given the important role they play, some oncogenes are used as biomarkers or are targeted by drugs to inhibit cancer progression.

Several molecular processes also contribute to carcinogenesis or irregularities in the gene expression levels of cancer cells:

- Modifications of the karyotype (number and appearance of the cell chromosomes), might lead the cell to an aneuploid state which contributes to cancer progression [37]. Some regions (or even complete chromosomes) might be copied or deleted.
- Expression of miRNAs play a regulatory role for other genes [31]. The miRNAs are a type of small non-coding RNAs that bind to the transcripts of other genes, thus preventing their translation to proteins.
- Methylation is an epigenetic process that also plays a role silencing certain key genes [42]. Tumor-suppressor genes might be affected in this way.

Cancer types can be classified by tissue of origin, e.g. stomach, colon, breast, among others. But finer classifications could exist inside these types.

2.2 Breast Cancer Classification

Breast cancer develops from breast tissue. Breast tissue includes the one from the lobules (glands that make milk), the ducts (thin tubes that carry the milk from the lobules to the nipples), lymph nodes and blood vessels. Breast cancer usually originates from the tissue of the ducts (ductal carcinoma) or from the lobules (lobular carcinoma), but could also originate from other places. If the abnormal cells haven't spread to neighboring tissues, then it is said that the cancer is *in situ*. Otherwise it is called breast invasive carcinoma [27]. Breast cancer is the most common cancer on woman. It is estimated that it affects about 12% women worldwide [24].

Breast cancers can be categorized using a variety of criteria. Every one of them provides some input that can be used by oncologist to determine the best course of action when treating a patient:

1. Histopathology: refers to an assessment made by microscopic observation of tissue. As described previously, common classifications in this regard are ductal carcinoma

and lobular carcinoma (depending of the primary site of the tumor). Tumors in the same histopathological category can follow different clinical courses and show different responses to the same therapy [15, 6]. This reveals the necessity of other means of classification.

2. Grade: indicates how much have cancer cells lose their differentiated state in comparison to normal cells of the tissue. Common descriptions are well differentiated (low-grade), moderately differentiated (intermediate-grade) and poorly differentiated (high-grade). The higher the grade, the worse the prognosis. It is also expected that higher grade tumors will differ in a greater way from normal cells in their gene expression levels.

3. Stage: the relative size of the tumor and how much has spread to neighboring tissues. Stage 0 refers to *in situ* state (benign tumor). Stages 1-3 indicate an invasive state and the higher the number, the greater the size of the tumor and dispersion within the breast or lymph nodes. Stage 4 indicates metastatic state. Again, higher numbers have associated a less favorable prognosis.

4. Receptor status: in breast cancer cells there are 3 receptors of particular importance: estrogen receptor (ER), progesterone receptor (PR) and human epidermal growth factor receptor 2 (HER2). Their presence or absence is currently an important input for deciding the most appropriate therapy. For example, estrogen receptor positive cancer cells (ER+) need estrogen in order to growth, so they can be treated in a way that specifically target estrogen levels. Triple negative cancers (ER-/PR-/HER2-) currently have the worse prognosis because of the lack of targeted treatments.

5. Molecular subtypes: these types give an insight into the receptor status along with tumor grade and prognosis [20]. Novel approaches might categorize cases into one of the 4 main classes:

   - Luminal A: Tend to be estrogen receptor-positive (ER+), HER2 receptor-negative (HER2-) and tumor grade 1 or 2. They also tend to have the best prognosis and can be treated with hormone therapy. Most breast tumors are Luminal A.
   - Luminal B: Tend to be estrogen receptor-positive (ER+), may be HER2 negative or positive (HER2-/HER2+). Have poorer prognosis than luminal A reflected in aspects like a higher grade and stage.
   - Basal-like: The majority of these cases are triple negative (ER-/PR-/HER2-). Of all subtypes, these have the poorer prognosis and tend to be aggressive. These are usually treated with surgery, radiation therapy and chemotherapy.
   - HER2-enriched: Most of them (70%) are HER2+. These also tend to be ER- and PR- and have a poorer tumor grade. HER2 type breast cancers can be treated with anti-HER2 drugs like trastuzumab (Herceptin).

   There has been great interest in characterizing these subtypes at the molecular level [44].

2.3 Pattern Recognition and Artificial Neural Networks

Pattern recognition is the act of taking in raw data and taking an action based on the category of that pattern [10]. A common pipeline followed on the Pattern Recognition paradigm involves the steps of data preprocessing, feature extraction, and then, classification (usually performed by some previously trained model). A lot of other challenges and sub-problems are usually present when building pattern recognition systems, among these: noise in the data, overfitting, high dimensionality, use of prior knowledge, missing features and others.

If the samples from the training data possess labels that are used by the learning algorithm then the process is said to be supervised. Otherwise it is said to be unsupervised. Unsupervised learning is popular for performing exploratory data analysis where a clear structure of data is not known beforehand. It serves well for finding candidate categories for the data and identifying features of interest [10]. Major departures from expected characteristics (like the discovery of subclasses) can be found when using unsupervised learning methods.

Clustering is probably the most common task under unsupervised learning. The idea behind consists of finding a partition of a set of entities, such that similar ones are combined in the same clusters, while dissimilar entities end up on different clusters [26]. The concept of similarity might vary entirely from one clustering algorithm/methodology to another. Hence, the problem of finding the best clustering algorithm for a given problem is not straightforward.

Artificial neural networks are computing systems that seek to mimic the computational capabilities present in the biological neurons [36]. These systems are used for both supervised and unsupervised learning tasks. An artificial neural network has a collection of connected computation units. These connections (usually weighted) mark the directionality of signals between computing units. If the amount of input signal overpass some threshold, then the given unit triggers a signal that is sent to other connected units. Flows of signals in this fashion (along with the correct weights between units) allow the computation of functions. Learning a function with an artificial network usually involves adjusting these weights.

One of the categorizations used for artificial neural networks takes into account how the computation units are connected. The most common type, feed-forward neural networks don't posses cycles between units and their connections. On the other hand, recurrent neural networks might have connections between units that form cycles.

2.4 Hopfield Networks

A Hopfield network is a type of recurrent neural network with n computing units, where n is also the number of dimensions of the data. Each computing unit acts as an input and

output unit, which results in a matrix W of approximately $n^2/2$ weighted links representing the network (basically, a complete undirected graph). See figure 2.1.

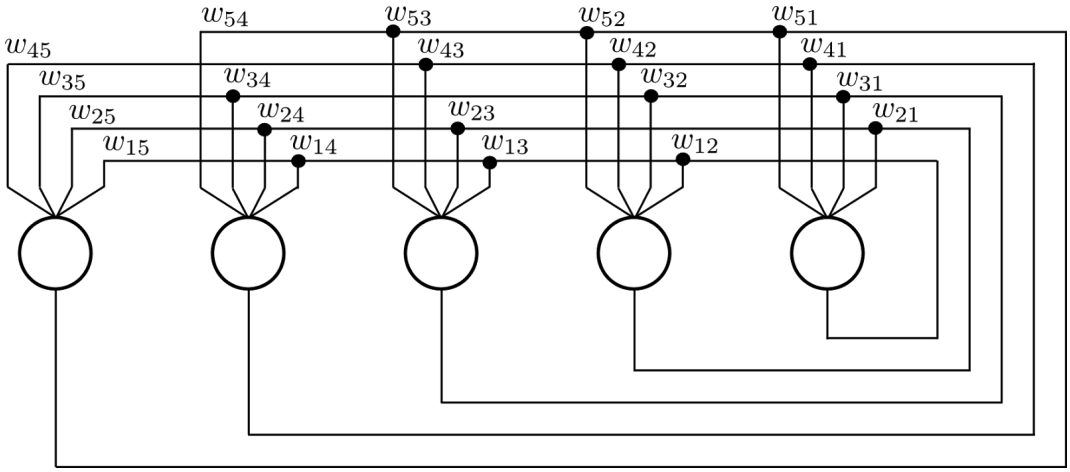


Figure 2.1: A Hopfield Network is a full, undirected graph. In other words the weight w_{nm} is equal to the weight w_{mn}. This property is necessary for the network to converge to a stable state when "recalling" a stored pattern from a input pattern.

Two popular approaches for calculating the weights between links are Hebbian learning and Pearson correlation indexes. The network can be trained using Hebbian learning with the following formula:

$$W = \frac{1}{m} \sum_{k}^{m} p_k p_k^T - I \tag{2.1}$$

Here, m is the number of patterns used to train the network. The formula calculates the sum over the outer products of the pattern vectors p_k, then normalizes values and sets the diagonal with zero values. The result, the matrix W, contains the numerical values of the network links, where W_{ij} is the weight of link connecting computing unit i to computing unit j.

On the other hand. It is possible calculate the weights of the matrix W as Pearson correlation indexes. The entry W_{ij} corresponds to the Pearson correlation index between the values of feature i with that of feature j.

After being trained, the network can be used to "recall". This means that given a new input pattern p, the network can associate the pattern p with the closest pattern q that it remembers. This ability to reconstruct stored patterns from similar input is the reason Hopfield networks are considered a way of implementing associative memory. Input patterns associated with the same stored pattern form a cluster of similar entities.

The "recall" phase for a input pattern works as follow. The process is divided into different

time steps t. At any time step t', each computing unit has a binary state $s_i \in \{-1, 1\}$, $i \in 1, \ldots, n$. The states for the next time step are calculated as follows:

$$s_i^{(t'+1)} = sgn\left(\sum_j^n W_{ij} s_i^{(t')}\right) \tag{2.2}$$

The sgn is defined in the following way:

$$sgn(x) = \begin{cases} +1 & x > 0 \\ -1 & x \leq 0 \end{cases} \tag{2.3}$$

Given an input pattern $p = (p_1, \ldots, p_n)$ for the "recall" phase, the initial state of the network is calculated as $s^0 = (sgn(p_1), \ldots, sgn(p_n))$. Then, equation 2.2 is run for the t iterations. It is guaranteed that if $t \to \infty$, then the state of the network will converge to a stable state with minimum energy.

If all computing units are updated simultaneously (like described previously) then the model it is said to be synchronous. Otherwise, if computing units are updated one at a time then the model is asynchronous. Some authors argue that the asynchronous model better mirrors the biological nature of how neurons work.

Chapter 3

Methods

The present chapter purpose is to fully describe the methods that were followed in order to fulfill each one of the specific objectives.

3.1 Computing Platform

As a methodological aspect relevant across all specific objectives, it is stated that the software platform used for running algorithms, visualizing data and performing analysis was exclusively the R Statistical Computing Environment version 3.4.3 (2017-11-30) – "Kite-Eating Tree". It is also worth mentioning that the IDE RStudio (Version 1.1.414) was used in order to facilitate the coding process. Its debugging and documentation features were worth exploring and contributed to the development work.

3.1.1 Used Packages

Relevant used R packages[1] (and their respective versions) are listed below:

- **TCGAbiolinks** (2.6.12) - **An R/Bioconductor package for integrative analysis with GDC data**: Provides facilities for querying and downloading data from the Genomic Data Commons repository. It also includes other convenient features, namely, the capacity of loading downloaded data as R data frames (table-like data structures) and the access to premade datasets with annotation labels for the TCGA patients.

[1]The Bioconductor version used was 3.6. Every **Bioconductor** package is also a R package, but it might follow a different release cycle that differs from the normal R scheduled releases.

- **CNTools** (1.34.0) - **Convert segment data into a region by sample matrix to allow for other high level computational analyses**: This Bioconductor package allows transforming copy number per segment data to a copy number per gene matrix. This is important in terms of compatibility with other data types, such as expression data which is given in terms of genes (not regions).

- **Matrix** (1.2-12) - **Sparse and Dense Matrix Classes and Methods**: This package facilitates more efficient representations and operations for matrix structures than the native ones from R.

- **scatterplot3d** (0.3-41) - **3D Scatter Plot** and **rgl** (0.99.16) - **3D Visualization Using OpenGL**: Used for supporting and enhancing the visualization of data, specifically after performing Principal Component Analysis.

- **gglot2** (2.2.1) - **Create Elegant Data Visualisations Using the Grammar of Graphics**: Plotting library for generating the graphics in the "Results" Chapter.

- **ClusterR** (1.1.1) - **Gaussian Mixture Models, K-Means, Mini-Batch-Kmeans and K-Medoids Clustering**: Mainly used because it provides functions for calculating clustering validation metrics like Rand Index, Adjusted Rand Index, Precision, Recall, among others.

- **dbscan** (1.1-2) - **Density Based Clustering of Applications with Noise (DBSCAN) and Related**: Contains implementations for the DBSCAN and OPTICS clustering algorithms. OPTICS was used in the experimental runs in order to have a reference point, hence not relying exclusively on the Hopfield Network's performance for drawing conclusions about multi-omic data usage.

- **FactoMineR** (1.41) - **Multivariate Exploratory Data Analysis and Data Mining**: Used for its more sophisticated PCA plotting functions.

3.2 Source Code Organization

R is an interpreted language. As such it provides very good prototyping capabilities (line by line execution, relaxed typing system, writing complex tasks with a minimum of code). However, the developer must beware relying too much on these features, as the code can easily reach an unmaintainable state and have poor modularization qualities. These reasons suggested the necessity of organizing the code among different project units and source files. This basic organization can be easily reused for other data-intensive kind of projects.

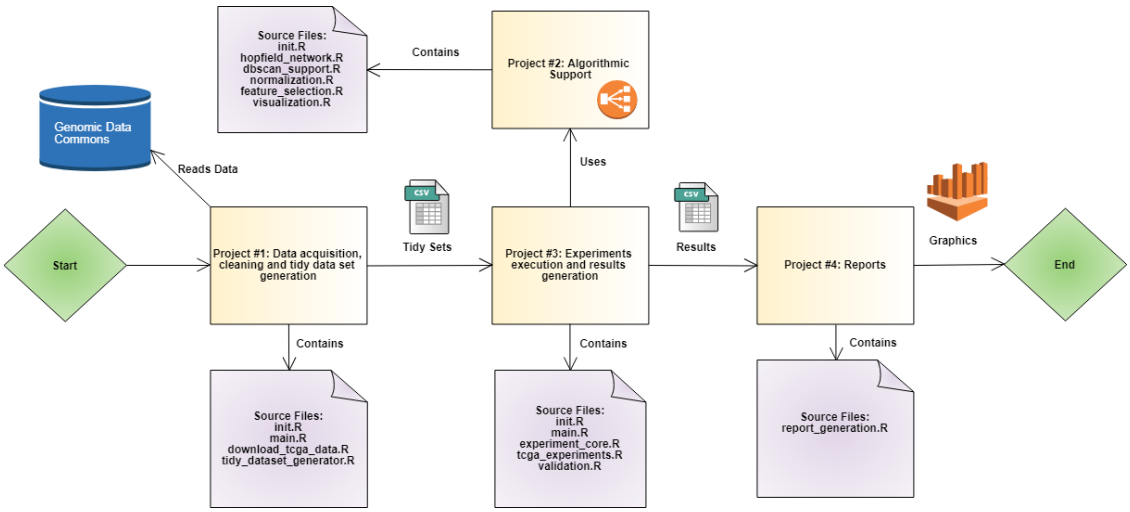


Figure 3.1: Projects organization and dependencies.

3.2.1 Projects

Projects provide the highest level of organization for source code. These reflect the following processes:

1. Data acquisition, cleaning and tidy dataset generation: This project contains all code related to using the GDC client tool for downloading the required data, cleaning it and generating datasets that are ready to be processed by algorithms.

2. Algorithmic support: Contains code implementing machine learning, normalization and feature selection algorithms.

3. Experiments execution and results generation: Provides the infrastructure for running experiments in an efficient and organized way. This project depends on "Algorithmic support" and generates results datasets that act as the input for the "Reports" project.

4. Reports: Generates graphics summarizing the results obtained by previously running the experiments.

As can be seen, these projects constitute a natural organization of code that reflects the pipeline through which data is transmitted from one unit of organization to another. See figure 3.1.

Each project follows the basic folder structure showed on figure 3.2.

In the following sections, more detail is provided about the different processes supported by these project entities.

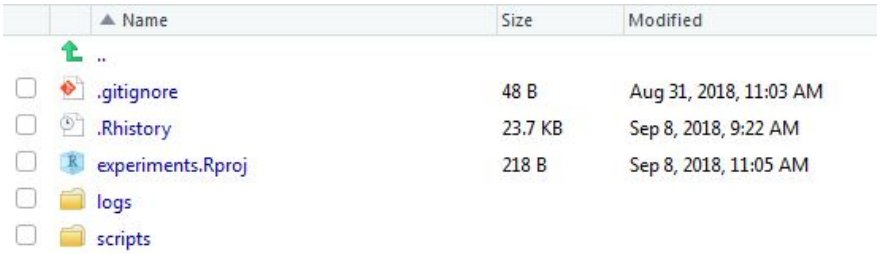


Figure 3.2: Basic project structure. a) *.Rproj* file contains basic configuration for the project. b) *logs* folder collects the log files generated after the *main.R* script is executed. c) *scripts* folder contains all *.R* files associated with this project. There are two mandatory files on most of the projects: *init.R* for installing missing package dependencies and *main.R* which launches the process associated with the project in order to generate the output files. d) A git repository is also defined at this level for source control purposes.

3.3 Data Download and Tidy Dataset Generation

3.3.1 About The Cancer Genome Atlas Data Source

The Cancer Genome Atlas is now part of the Genomic Data Commons project. In summary, the Genomic Data Commons is a property oriented graph database which organizes a great diversity of files. Information could be classified as genomic, clinical or biospecimen depending of its nature. Genomic data categories include: sequencing data, copy number variation, DNA methylation, simple nucleotide variation, genomic profiling and transcriptome profiling.

Each data category uses its own data file formats (for example .tsv for transcriptome profiling and .bam for aligned reads). Also each data category can be further subdivided into different data types. Transcriptome profiling has three subtypes e.g: gene (mRNA) expression, exon expression and miRNA expression. For this study three data types for each cancer case were used: gene (mRNA) expression and miRNA expression (both belonging to the transcriptome profiling category), and the masked copy number segment data type (which belongs to the copy number variation category).

Some data categories and types are showed in the table 3.1, just to emphasize the difference between them and the great diversity of multi-omic data available in the Genomic Data Commons repository.

3.3.2 Data Download

The files from the Genomic Data Commons repository can be accessed in mainly three ways:

Data Type	Data Category	File Extension	Experimental Strategy
Gene Expression Quantification	Transcriptome Profiling	.tsv	RNA-Seq
miRNA Expression Quantification	Transcriptome Profiling	.tsv	miRNA-Seq
Masked Copy Number Segment	Copy Number Variation	.tsv	Genotyping Array / Targeted Sequencing
Methylation Beta Value	DNA Methylation	.tsv	Methylation Array
Aligned Reads	Sequencing Data	.bam	RNA-Seq / miRNA-Seq
Raw Simple Somatic Mutation	Simple Nucleotide Variation	.vcf	Various
Simple Germline Variation	Simple Nucleotide Variation	.vcf	Various

Table 3.1: Examples of data types supported by the6 Genomic Data Commons repository.

1. Using the GDC Data Portal: files to download are selected using a web user interface in a style similar to e-shopping. See https://portal.gdc.cancer.gov/.
2. Using the GDC Data Transfer Tool: a standard client-based mechanism supporting high performance data downloads and submission.
3. Directly consuming a public RESTful API.

The first option was immediately discarded, as some automated way of downloading data is much more preferred over the manual selection of files (at least for the volume intended to be downloaded which summed up to 3000 files approximately). There was one file per data type per patient.

Given that the implementation platform for this project was R, two Bioconductor packages that consume the GDC RESTful API seemed to be good options for downloading and preprocessing the necessary data. These were the "TCGABiolinks" and "GenomicDataCommons" packages. The second one is currently maintained by an official member of the National Institute of Health and its API really mirrors the web service structure and the resources it exposes, so it was considered as the first implementation option. However, this package lacked a crucial feature, that was, the capacity of merging all the files for the same data category into a single R dataset.

On the other hand, the "TCGABiolinks" package supported loading the data more easily as R/Bioconductor objects and in general had better preprocessing capabilities. Another relevant feature of this package (even if not used for this project), is the capacity of populating "SummarizedExperiment" Bioconductor objects from the downloaded data. These possess facilities for organizing metadata of patients and features (genes).

After these considerations, the "TCGABiolinks" package was used in conjunction with the GDC Data Transfer Tool (which can be invoked from R). Basically three steps are required by the package to download a group of files and merging them into a single dataset. These are:

1. Create a GDCQuery object with the proper parameters, like in the following example:

```
GDCquery(project = "TCGA-BRCA",
data.category = "Transcriptome Profiling",
data.type = "Gene Expression Quantification",
workflow.type = "HTSeq - FPKM")
```

   The call to the function **GDCquery** returns an object containing metadata of the files to download (size, data category and type, among other details) but it does not execute the actual download process. This is useful, as some rechecking can be done, with the possibility of excluding some files from the final download list. The previous query selects cases from the "TCGA-BRCA" (that is Breast Invasive Carcinoma) group, from which it retrieves gene (mRNA) expression quantification files (belonging to the transcriptome profiling category) and which are normalized using the FPKM (Fragments Per Kilobase) procedure [45].

   Using raw RNA-Seq reads is not recommended as the gene length (number of base pairs) might introduce a bias. Longer genes capture more reads which does not necessarily translate in higher transcription rates [35]. The FPKM normalizes expression rates according to each gene's length.

2. Call the GDCdownload function with the GDCQuery object:

```
GDCdownload(query = query.object, method = "client")
```

   The parameter "method = client" indicates that the "gdc-client.exe" executable is used for performing the download (which can be previously acquired through the GDC's website). Using the client application is in general a more efficient and robust method for downloading a high quantity of files. The default method uses direct HTTP requests to the REST service which is better suited for a small amount of files.

3. Call the GDCprepare function with the GDCQuery object:

```
data <- GDCprepare(query = query.object,
summarizedExperiment = FALSE)
```

   This function will look for the previously downloaded files and will merge them into a single R data frame (which still requires some cleaning described in the next section). The resulting dataset will of course vary in their variables depending of the processed

data type. However all share the same observations (breast cancer patients samples in this case).

The previous process was repeated for each one of the three kinds of genomic data types that were used: gene (mRNA) expression, miRNA expression and masked copy number segment, varying the "GDCQuery" accordingly.

3.3.3 Tidy Dataset Generation

The generated sets still required some extra processing before they are ready to be used. The process of "tidying" refers to cleaning the data from unwanted values and restructuring the observations and variables so they are easy to manipulate [48]. This process is somewhat similar to data normalization on relational databases. Fortunately the tidying process was minimum. This is described in the following sections.

Previous Steps to Generation of Tidy Sets

As a prerequisite to the process of generating the tidy sets for each one of the molecular types used, it was necessary to generate two auxiliary datasets.

1. The first of them contains the breast cancer type labels for each sample, that is the type of breast cancer for each patient (luminal A, luminal B, HER2 or basal). The package "TCGABiolinks" provides a function **PanCancerAtlas_subtypes** that returns a predefined dataset with this information. A set of filters were applied to the returned set.

   (a) First, filtering the cases to the TCGA-BRCA group of samples.

   (b) Second, restricting the type of tissues to solid tumors. This must be deduced from the structure of the TCGA bar codes. These are uniques identifiers used to recognize one sample from another inside the TCGA database. They codify several information pieces separated by hyphen characters. One of these segments indicates the type of tissue. Other type of tissues include "normal-like", whose most of their cells are still in their normal differentiated states. Some patients have samples for both normal and solid tumor states, but only the latter ones were kept.

   This dataset was used for the generation of other tidy sets and for clustering validation purposes as well.

2. The second auxiliary set contains the list of all the protein coding genes. This can be generated by a custom tool (genome browser from the www.ensembl.org website). This set was used for filtering purposes.

Gene (mRNA) Expression Data

Taking a quick look at how data was organized, it first came to attention that the data appeared to be transposed. That is, patients corresponded to columns and genes to rows. For the ease of treatment it was kept in that way unless some function required the opposite format. This clarification is necessary, as the variables in this study might be referred as the dataset rows (instead of columns) in the following sections. This also applies to the columns, which might be referred as the cases.

For the gene (mRNA) expression data the following changes were applied:

1. Assigning values of the first column to be the "row names" of the dataset. These values corresponded to gene ids as they appear in the Ensembl database. After this, the column was eliminated.

2. Bar codes were cut out to the first 16 characters. The other bar code segments contained details that were not useful for the study.

3. Patients for whose breast cancer subtype are not in the "Labels" auxiliary dataset are removed.

4. The second auxiliary set was used to limit the genes to only protein coding genes (around 20000). This constitutes a 3:1 ratio reduction, as the original number of genes is around 60000. The total group of genes include pseudo-genes or genes that produce some kind of non-coding RNA.

See figure 3.3 for the resulting dataset.

miRNA Expression Data

The nature of miRNA Expression data is very similar to the more conventional gene (mRNA) expression data (a numeric matrix of cases versus genes). The primordial difference is that GDC files related to this type of data contain both raw and normalized values (with the reads per million mapped reads procedure, see figure 3.4), whereas the gene expression data divides both types of values into different kind of files (hence only files for the normalized ones were retrieved). The tidy dataset generation procedure was as follows:

		TCGA-E2-A1BC-01A	TCGA-AC-A8OR-01A	TCGA-AO-A0JJ-01A	TCGA-E2-A14N-01A	TCGA-GM-A4E0-01A	TCGA-A7-A5ZX-01A	TCGA-BH-A1EO-01A
1	ENSG00000000003.13	4.447345	4.39790485	14.179096	11.9604598	11.7321079	22.755259	17.6934675
2	ENSG00000000005.5	1.354929	0.01619332	1.919527	0.1371534	9.3183301	0.161233	0.3788116
3	ENSG00000000419.11	26.539069	23.37123574	26.448078	40.5809657	23.2858085	32.789360	27.2198268
4	ENSG00000000457.12	6.388660	3.21203130	4.885382	2.7327609	2.9613772	6.405571	5.8673697
5	ENSG00000000460.15	1.218611	1.03987851	1.877466	4.1077069	0.9706903	1.699982	2.8485465
6	ENSG00000000938.11	2.027008	1.38836482	3.462328	2.1784689	14.6319212	2.178442	3.9614927
7	ENSG00000000971.14	9.021477	0.54735200	9.037279	1.3013157	10.3326775	5.376308	13.6246607
8	ENSG00000001036.12	12.236241	31.58792543	23.066637	21.9861083	21.8938336	14.977856	15.1239823
9	ENSG00000001084.9	4.942689	8.78899295	4.124768	3.8687391	4.5589517	4.258821	3.9134467

Figure 3.3: Resulting gene expression dataset which contains Fragments Per Kilobase of transcript per Million mapped reads.

	miRNA_ID	read_count_TCGA-E2-A10E-01A-21R-A10I-13	reads_per_million_miRNA_mapped_TCGA-E2-A10E-01A-21R-A10I-13	cross-mapped_TCGA-E2-A10E-01A-21R-A10I-13
1	hsa-let-7a-1	39515	22750.1320	N
2	hsa-let-7a-2	39497	22739.7688	N
3	hsa-let-7a-3	39430	22701.1946	N
4	hsa-let-7b	147144	84715.8148	N
5	hsa-let-7c	3285	1891.2864	N
6	hsa-let-7d	803	462.3145	N
7	hsa-let-7e	2293	1320.1582	N
8	hsa-let-7f-1	6817	3924.7792	N
9	hsa-let-7f-2	6769	3897.1440	N

Figure 3.4: Raw miRNA expression data (as generated by **GDCPrepare** function). Each sample contains three associated columns: **read count**, **read per million miRNA mapped** and **cross-mapped**. Only one sample's data is showed because of space limitations.

1. All columns with raw count values and cross mapped flags were eliminated.
2. The remaining ones were renamed so they only contained the TCGA bar code for the samples.
3. Bar codes were cut out to the first 16 characters.
4. Patients for whose breast cancer subtype are not in the "Labels" auxiliary dataset are removed.

See figure 3.5 for the resulting dataset.

Copy Number Data

TCGA Copy number data is organized into genomic segments. That is, the dataset is composed of one record per affected region (i.e showing copy number variation), potentially

		TCGA-E2-A10E-01A	TCGA-A7-A26G-01A	TCGA-A8-A06Q-01A	TCGA-A2-A0ES-01A	TCGA-S3-AA15-01A	TCGA-A8-A09Q-01A	TCGA-BH-A1FC-01A	TCGA-BH-A1EW-01A
1	hsa-let-7a-1	22750.1320	8059.5406	11922.7352	8542.1361	7217.4258	8297.5174	12338.5894	16382.3286
2	hsa-let-7a-2	22739.7688	7899.7393	11778.7802	8456.4323	7114.9488	8361.1378	12274.4906	16443.2121
3	hsa-let-7a-3	22701.1946	8081.7738	11803.6000	8571.6670	7310.3890	8318.7242	12476.4459	16563.1885
4	hsa-let-7b	84715.8148	22710.7662	21513.1159	35586.9910	11749.2472	20394.5880	6337.8835	22997.5045
5	hsa-let-7c	1891.2864	3508.2161	336.8405	5975.1940	2399.6473	523.8082	765.6739	4626.7911
6	hsa-let-7d	462.3145	823.0922	604.8946	573.2848	424.8583	554.9115	3366.5067	247.4737
7	hsa-let-7e	1320.1582	1111.6608	891.3863	674.3961	1041.8949	1567.8901	1483.9323	2343.3001
8	hsa-let-7f-1	3924.7792	2276.5886	1710.4406	1344.9403	2833.7475	1735.4239	12515.9589	6292.8513
9	hsa-let-7f-2	3897.1440	2349.7729	1669.3106	1352.6440	2904.1495	1820.9580	12600.2532	6295.3582

Figure 3.5: Resulting miRNA dataset which only contains read per millions mapped reads values.

	Sample	Chromosome	Start	End	Num_Probes	Segment_Mean
1	TCGA-EW-A1IW-10A-01D-A13N-01	1	3301765	247650984	129998	0.0051
2	TCGA-EW-A1IW-10A-01D-A13N-01	2	480597	241537572	132337	0.0008
3	TCGA-EW-A1IW-10A-01D-A13N-01	3	2170634	197812401	107297	0.0056
4	TCGA-EW-A1IW-10A-01D-A13N-01	4	1059384	107713424	60057	0.0028
5	TCGA-EW-A1IW-10A-01D-A13N-01	4	107714338	107749370	10	-0.6135
6	TCGA-EW-A1IW-10A-01D-A13N-01	4	107756802	187842528	43569	0.0050
7	TCGA-EW-A1IW-10A-01D-A13N-01	5	913983	180934240	101251	0.0033
8	TCGA-EW-A1IW-10A-01D-A13N-01	6	1011760	170596889	97296	0.0027
9	TCGA-EW-A1IW-10A-01D-A13N-01	7	664936	158592540	82102	0.0043
10	TCGA-EW-A1IW-10A-01D-A13N-01	8	667625	144182542	82283	0.0054
11	TCGA-EW-A1IW-10A-01D-A13N-01	9	789794	138044505	68614	0.0037
12	TCGA-EW-A1IW-10A-01D-A13N-01	10	366509	133411599	81464	0.0025

Figure 3.6: TCGA copy number segment raw data (as generated by **GDCPrepare** function). **Sample** corresponds to the sample's barcode which is an unique ID. The **Start** and **End** variables denote the base coordinates of the affected region. **Segment_Mean** is a value calculated from the copy number of a region with the formula: $log2(copynumber/2)$, so diploid regions have a segment mean of zero, amplified regions have a positive value and deletions a negative one.

having several records for the same sample, regions containing more than one gene, or genes being split into different regions. See figure 3.6. This format is not suited for high-level analyses like clustering, neither for integration with expression data which uses a matrix of genes by samples. A transformation to such format was necessary in order to perform data integration. Fortunately, the **CNTools** package provided the means to perform the task.

The R code for generating a reduced segment dataset[2] is showed in figure 3.7. First a **CNSeg** object is created by passing the raw **segments.dataset** as argument. Then the function **getRS** does the heavy transformation work. Important arguments are:

1. **by**: indicates the resulting features, in this case genes.
2. **what**: value to use when more than one segment includes the same gene. Mean values

[2]This is a term used by the creators of **CNTools**, which refers to the resulting dataset that has a single measurement per sample/feature combination.

```
seg <- CNSeg(segList = segments.dataset)
rsByGene <- getRS(seg, by = "gene", what = "mean", geneMap = gene.
    map.dataset)
result <- rs(rsByGene)
write.csv(x = result, file = output.path)
```

Figure 3.7: Code for generating a segment reduced set with genes as features for each sample.

	TCGA-3C-AAAU-01A	TCGA-3C-AALI-01A	TCGA-3C-AALJ-01A	TCGA-3C-AALK-01A	TCGA-4H-AAAK-01A	TCGA-5L-AAT0-01A	TCGA-5L-AAT1-01A	TCGA-5T-A9QA-01A	TCGA-A1-A0SD-01A
ENSG00000160752	0.14720	0.2089	0.3198	0.3681	0.0002	0.3560	0.2701	0.7612	0.6729
ENSG00000131910	0.14790	-0.4015	-0.2087	-0.0234	-0.0260	0.0211	0.0074	-0.5435	-0.3873
ENSG00000067334	0.26130	-0.3275	0.0736	-0.0234	0.0061	-0.0016	0.0065	-0.4575	0.0659
ENSG00000011007	0.14790	-0.4015	-0.2087	-0.0234	-0.0260	0.0211	0.0074	-0.5435	-0.3873
ENSG00000160691	0.81050	0.2089	0.3198	0.3681	0.0002	0.3560	0.2701	0.7612	0.6729
ENSG00000117318	0.14790	-0.4015	-0.2087	-0.0234	-0.0260	0.0211	0.0074	-0.5435	-0.3873
ENSG00000143578	0.81050	0.2089	0.3198	0.3671	0.0002	0.3560	0.2701	0.7612	0.6729
ENSG00000143556	0.18390	0.2819	0.3198	0.3671	0.0225	0.3560	0.2701	0.7612	0.6729
ENSG00000163221	0.18390	0.2819	0.3198	0.3671	0.0225	0.3560	0.2701	0.7612	0.6729

Figure 3.8: The resulting gene copy number dataset. Has one value for each sample/gene combination.

were used for this case.

3. **geneMap**: a dataset that contains the base coordinates and chromosome number for each gene. A custom set was generated using the BioMart online tool using the GRCh38 genome version, which is the same used by the Genomic Data Commons datasets.

The generated set is extracted using the **rs** function and then persisted to a .csv file. Still, some cleaning steps similar to those applied to the expression datasets were necessary, namely: reassigning row names, keeping only protein coding genes, modifying barcodes and eliminating unnecessary variables (like base coordinates and chromosome number). After these steps were effectuated, the gene copy number dataset was in tidy form and ready for being used. See figure 3.8.

3.3.4 Extra Normalization Steps

The distributions of values for gene (mRNA) expression variables were found to be highly skewed (differences between gene expression levels were sometimes exponential), so log2 transformations were applied for the ease of treatment and visualization. On the other hand, miRNA expression variables did not need this transformation.

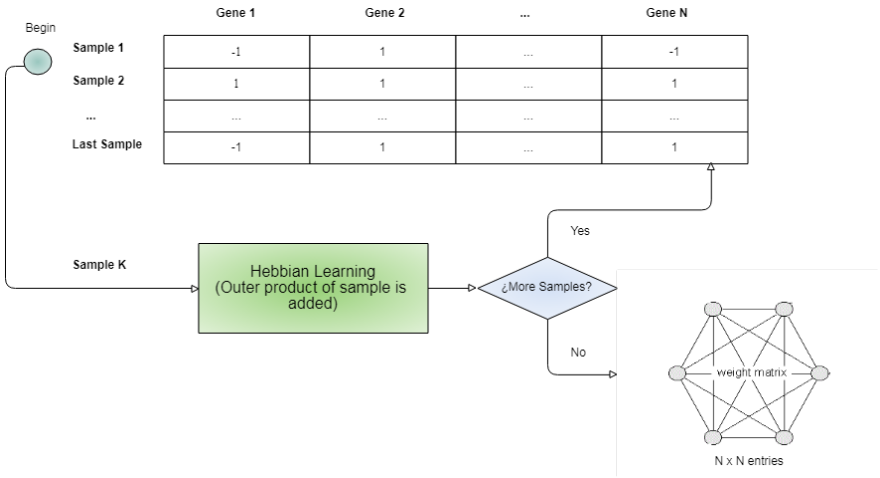


Figure 3.9: High level view of the training procedure.

Both gene (mRNA) and miRNA Expression variables were z-normalized, a step necessary so they could be integrated together under the same data table.

3.4 Implementation and Visualization of Hopfield Model

3.4.1 Training and Recall Procedures

According to the method used by [22], using the Hopfield Network as a clustering procedure requires training the network (using the Hebbian rule) with each sample's binarized vector[3] (whose dimension is equal to the number of used features). Inputs can be binarized using a **sign** function which assigns 1 to values higher than 0 and -1 otherwise. See the figure 3.9 for a high level view of the training procedure.

Then the recall function is applied to each one of those same vectors, where inputs that converge to the same attractor state are considered to belong to the same cluster. See also figure 3.10.

3.4.2 Implementation Details

For implementation purposes, an example code from the course "Foundations of Neural and Cognitive Modeling" of the University of Amsterdam was used as base[4]. Logic is split into

[3]In this case the term binarized refers to having only two types of values (not strictly 0 and 1). For the present context values are -1 and 1, indeed. Hope this clarification will avoid future misunderstandings.

[4]http://www.illc.uva.nl/LaCo/clas/fncm13/assignments/computerlab-week4/.

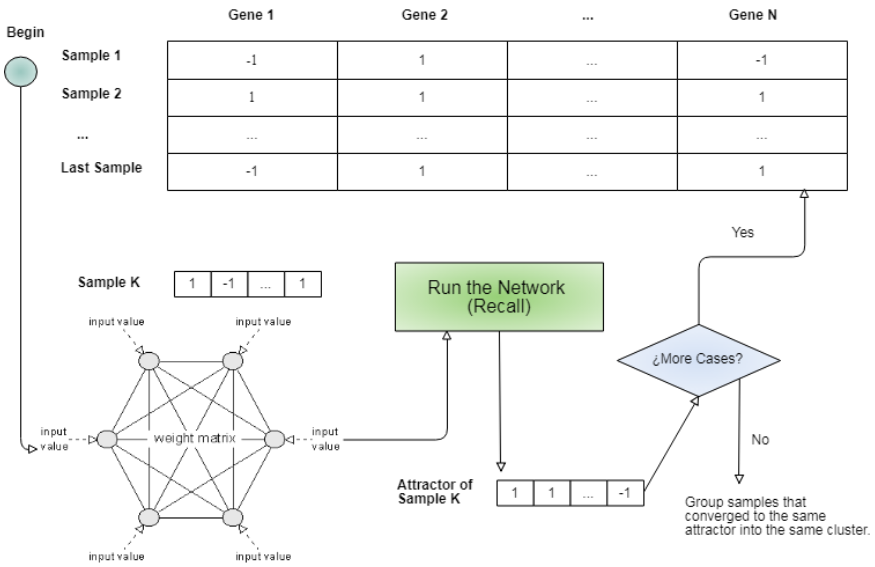


Figure 3.10: High level view of the recall procedure.

two different functions, one for training the network, i.e generating the matrix of weights, and the other for recalling memories (associating inputs with attractor states). Some minor modifications were made to the code. These included:

1. Substituting uses of base **matrix** R objects with **Matrix** S4 objects, more suited for higher computation rates and an efficient representation.

2. The outer product of the training equation 2.1 was calculated with the function **tcrossprod**, which overall is more efficient for this case.

3. The recalling function was modified to add the possibility of tracing convergence routes from the initial vector state to the final attractor state. This is done for visualization purposes and is described below.

The training function is very simple and relies mostly on linear algebra matrix operations. See figure 3.11. The function receives as a parameter a matrix of training inputs called **patterns**. It is assumed that these patterns are already in a binarized state (only have +1 or -1 values). The number of computing units (or neurons) N is determined by looking at the length of the first input pattern. Then the matrix of weights is initialized with dimensions N by N, and with a zero on each entry. The sum of outer products is calculated in a loop using the **tcrossprod** function. As the final steps, entries of the matrix's diagonal are set to 0 and values are normalized by the number of training patterns so each entry has a value between -1 and 1 (in a manner similar to most correlation coefficients).

The recalling function is showed in 3.12. It receives the following parameters:

```
create.hopfield.network.hebbian <- function(patterns) {
        number.of.neurons = length(patterns[[1]])
        weights <- Matrix(0, number.of.neurons, number.of.neurons)

        for (i in 1:length(patterns)) {
        weights <- weights + tcrossprod(Matrix(patterns[[i]]))
        }
        diag(weights) <- 0
        weights <- weights / length(patterns)
        return(list(weights = weights))
}
```

Figure 3.11: Hopfield Network training function

1. **hopfield.network.object**: Contains the matrix of weights from a previously trained network.

2. **pattern**: The input pattern that will be transformed into one of the attractor states stored in the network.

3. **max.iterations**: The maximum number of iterations (neuron updates) to try. The needed number of iterations increases with the quantity of features. It is worth mentioning that the update rule implemented on the function is asynchronous (neurons are updated one at a time), so each iteration performs a single update.

4. **replace**: Describes the way neurons are selected when updating them. If the parameter is set to **FALSE**, all neurons need to be updated before proceeding to the next round of updates. Otherwise neurons will be picked completely at random. In practice, the last option showed higher possibilities of converging to spurious attractors (local minima), so it was avoided when performing the experiments.

5. **trace.route.mode**: A flag indicating whether to trace intermediate states for vector **y**. These can be used for visualization purposes.

The result of executing the function is a vector representing the attractor state to which the input converged. If the **trace.route.mode** flag was set, then the result also contains a sequence of intermediate states.

3.4.3 Visualization Method

One of the interesting modeling points of the Hopfield Network is the capacity of representing a landscape with physical properties similar to the ones exposed in Waddington model of

```
run.hopfield <- function(hopfield.network.object, pattern, max.
    iterations=100, replace=FALSE, trace.route.mode = FALSE) {
        y = pattern
        number.of.neurons = dim(hopfield.network.object$weights)
            [1]
        transitory.points <- NULL
        if (trace.route.mode) {
                transitory.points <- data.frame()
        }

        order = c()
        converge = FALSE

        for(it in 0:(max.iterations-1)) {
                if (it %% number.of.neurons == 0) {
                        if (converge == TRUE) {
                                print('reach a stable state!!!')
                                break
                        } else if (trace.route.mode) {
                                transitory.points <- rbind(
                                    transitory.points, c(y, energy(
                                    y, hopnet)))
                        }
                        order = sample(1:(number.of.neurons),
                            replace=replace)
                        converge = TRUE
                }

                i = order[it %% number.of.neurons + 1]
                yi.old = y[i]
                z = hopfield.network.object$weights[i,] %*% y
                y[i] = sign(z)
                if (yi.old != y[i]) {
                        converge = FALSE
                }
        }
        if (!trace.route.mode) {
                return(y)
        } else {
                return(list(attractor = y, transitory.points =
                    transitory.points))
        }
}
```

Figure 3.12: Hopfield Network recalling function

cellular differentiation. This is useful for identifying, at least in an intuitive way, the relative distance between attractor states and properties like the size of their basins of attraction, i.e how big is the influence each attractor exerts on nearby vectors.

In [22] a visualization method is proposed. It consists in performing a principal component analysis of the training samples, then using the first two principal components as the x-axis and y-axis of the plot. Detected attractor states are also projected alongside the cases. A third z-axis is also added, which corresponds to the energy levels of the cases (a Lyapunov function). Cases with low energy levels are closer to attractor states that those with high levels. Given a vector S, the energy level can be calculated with the following formula:

$$E(S) = -\frac{1}{2} S^T W S. \tag{3.1}$$

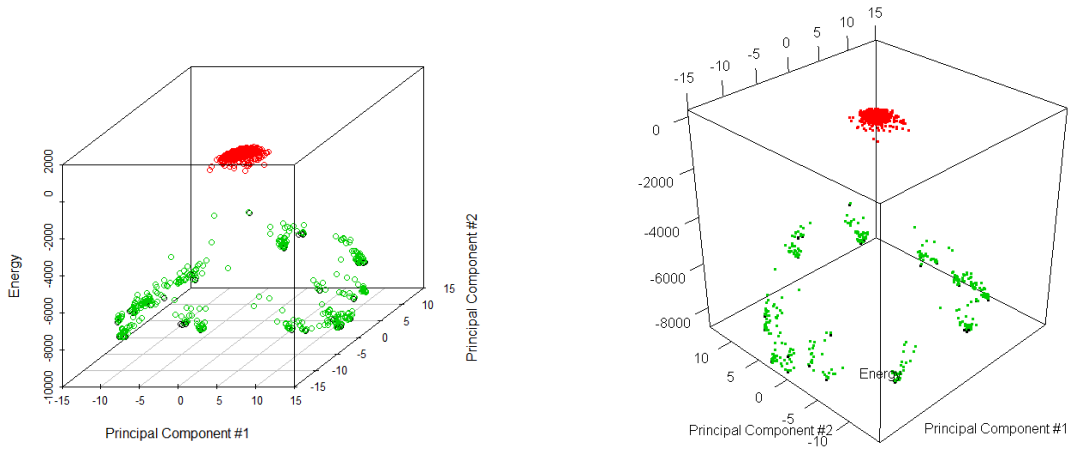
This function can be easily implemented in R with the following code. Note the use of the %*% operator for matrix multiplication:

```
energy <- function(point, network) {
        return (-0.5 * (as.numeric(t(point) %*% network$weights %*
          % point)))
}
```

The described method is useful for checking clustering results but does not necessarily reveal all attractor states in the network and their convergence paths. A monte carlo inspired visualization method that does not depend of a set of training samples was created as a support tool for this project.

The idea is fairly simple. A lot of random points are generated (binary vectors of size N) and these are recalled using the matrix of weights from the network that needs to be visualized. Transitory states for the points are captured as well. By using this randomized process, the network's attractors and their positions can be discovered.

The intuitive idea behind this method is the following. Imagine an object that physically exists in space (has volume and mass) but it cannot be seen for some reason. One could try to pour some painting on the object to reveal its contours and shape. This is analogous to generating a lot of random points and made them all converge to some attractor state. The trajectories followed by the points will always be downward (as the z-axis represents the energy, which becomes progressively lower as the vectors get near an attractor state). The end result is a drawing of the landscape. See figure 3.13 for an example. This method reinforces the notion that the Hopfield Network tends to cluster points around the perimeter of a circle in a symmetric way. Points separated by an angle of 180 degrees are complementary binary vectors, so it is safe to say that every attractor has a polar opposite.



(a) Visualization using **scatterplot3d** package.

(b) Visualization using **rgl** package.

Figure 3.13: Example of visualizing a Hopfield Network's landscape using the custom visualization method. The z-axis corresponds to the energy function. Red dots are initial states which are generated randomly with an uniform distribution. Green dots represent intermediate states and serve to trace possible convergence paths. Black dots are attractor states (either local minima or maxima). The visualized network corresponds to one that has stored a couple of binarized patterns for all even digits (2, 4, 6, 8, 0).

3.5 Experimental Design

The experiments were organized following a factorial design of $3 \cdot 4 \cdot 6$ (three different factors with three, four and six levels respectively). In a factorial design, runs are distributed equally across all the different experimental conditions that the levels of factors dictate (72 combinations on this case). This allows to later measure possible joint effects between independent variables into the dependent ones.

3.5.1 Independent Variables (Factors)

There were three different factors involved in the experiments: the genomic dataset, the clustering algorithm and the number of used features.

Genomic Dataset

The diversity of multi-omic data from the TCGA provides different data views for the same cancer samples. This opens up the door to performing interesting data integrations that might yield features with better discrimination characteristics than the ones from single datasets. The datasets (alongside its respective codes) for this factor were the following:

1. Gene (mRNA) Expression - GE: a dataset containing only the expression rates for the protein coding genes. These values are normalized using the FPKM (Fragments Per Kilobase of transcript per Million mapped reads) procedure.

2. Gene Copy Number - GCN: a dataset containing the copy number values of genes as $log2(copy - number/2)$. These are calculated as described in the "Tidy Dataset Generation" section.

3. miRNA Expression - MiRNA: this dataset contains the expression values for genes that code for miRNAs. These values are normalized under the RPM (Reads per Millions) procedure.

4. Gene (mRNA) Expression integrated with miRNA Expression - GE + MiRNA: an early integration approach is followed. That is, the variables of the two datasets are merged into one new dataset. A small minority of samples with mRNA measurements, but lacking the miRNA ones were discarded.

5. Gene (mRNA) Expression values divided by Gene Copy Number - GE + GCN: the values of this dataset are generated by taking each gene expression value and diving it by the corresponding raw gene copy number value. To obtain these raw values the inverse function of $log2(copy - number/2)$ is applied.

6. Gene (mRNA) Expression values divided by Gene Copy Number and then integrated with miRNA Expression - GE + GCN + MiRNA: this dataset is obtained by taking the previous one and then performing an early integration approach with the miRNA expression dataset.

Clustering Algorithm

Testing clustering algorithms other than the Hopfield Network it is important from a validation standpoint. Given the complicated noisy nature of gene expression data, any clustering algorithm might encounter problems establishing clear divisions between groups. So using other algorithms might give an insight of how much is noise in the data affecting the results, or, if one algorithm is clearly better than others for the type of data. Tested algorithms were:

1. Hopfield Network: the main algorithm of interest for the study. It does not need an initial number "k" of expected clusters. Theses are determined by the resulting attractors which act as representative elements of the cluster. It works on binary data only.

2. OPTICS: a density-based clustering algorithm similar to dbscan [2]. Works better with clusters of variable density, and similarly to Hopfield's method, does not require an initial "k". Given that the algorithm requires a cut point (similar to hierarchical

clustering), an heuristic was followed to determining such. Different cuts are tried in an incremental fashion until the outliers (points not assigned to any cluster) ratio is below 15%. The cut point is still increased until the established clusters begin to merge together. Previous to that moment, the cut is chosen and remaining points which don't belong to any cluster are assigned to the one which is nearest respectively.

3. K-means: a classic algorithm in clustering literature. It needs a value of "k", which was set to 4 (the number of known breast cancer types).

Number of Used Features

Features were chosen using a simple filtering method. These were ranked using BSS/WSS ratio (between-group sum of squares divided by within-group sum of squares). A way of rephrasing this it's by saying that features that have high variance between groups and low variance within groups generally have better discriminative power than other features. The groups on this case correspond to the four groups of breast cancer involved in the study. Once ranked, features were chosen by picking the ones with the top n values, which are 10, 25, 50 and 100 for this factor.

In general, the miRNA variables showed BSS/WSS ratios much smaller than the ones from mRNA variables. So rankings were applied separately in case the two types of variables were integrated together and a selection ratio of 20% was used for the miRNA variables, while the remaining 80% for the other type of variables. For example, if 100 features were used, 20 would be the top miRNA variables and the remaining 80 would be the top mRNA.

3.5.2 Dependent Variables

Excluding basic measurements like the **number of generated clusters** and **running time**, the dependent variables corresponded mostly to external validation measures, i.e., knowledge about the problem's domain was used for calculating these metrics. The PAM50 [50] breast cancer labels provided by the "TCGABiolinks" package were used to create a reference clustering that serves as groundtruth prior to each experimental run. It is composed of four clusters, one for each molecular subtype of breast cancer.

The dependent variables were the following:

Number of Generated Clusters

The number of generated clusters.

Running Time in Seconds

Wall clock time taken to execute the clustering algorithm. It does not include time to normalize values or choose features.

RAND Index

A symmetric measure of similarity between two clusterings C and C' in terms of much they agree when placing pairs of points into the same (or different) cluster. It ranges from -1 to 1, with positive values indicating a certain degree of agreement. It's important to notice that the clusterings don't need to have the exact same number of clusters in order to have an acceptable RAND index. As long as certain structure is preserved the results might be good. The index is calculated with the following formula:

$$R = \frac{a+b}{\binom{n}{2}}$$

Here the denominator is equal to the number of all possible pairs that can be made with the n elements. In the numerator we have the sum of two values: a which is equal to the number of pairs where both elements belong to the same cluster in both C and C', and b, the number of pairs where both elements belong to different clusters on both C and C'.

ARI (Adjusted RAND Index)

A corrected for chance version of RAND Index:

$$ARI = \frac{\text{Index} - \text{ExpectedIndex}}{\text{MaxIndex} - \text{ExpectedIndex}}$$

Basically it excludes the possibility of obtaining good scores by chance, so index values tend to be a lot lower compared to the plain RAND index.

Other Measures

These include precision, recall, sensitivity and specificity. See Annexes. Note that in the context of clustering, a true positive is translated in a pair where the two elements belong to the same cluster on both the result and groundtruth. On the other hand, a true false is

a pair where the elements belong to different clusters, and both the result and groudtruth agree on this. Note that recall and sensitivity are the same, but the same not applies to precision and specificity.

3.5.3 Conditions Across Runs

1. Memory clean up: The garbage collector of R is triggered before each run with the intend of normalizing the available memory across different runs.

2. Used cases: Stratified random sampling is performed in order to choose an equal number of cases (82) from each of the four subtypes of breast cancer.

3. Replicas: Each of the 72 experimental conditions had 10 different runs for a total of 720. Results were averaged.

4. PCA views: Principal Component Analysis plots are generated for each run, both for the actual results and the groundtruth clustering.

5. Normalization: Gene (mRNA) expression variables are log transformed and Z-score normalized before any of the runs. The log transformation is not applied to miRNA varibles or gene copy number values.

Chapter 4

Results and Discussion

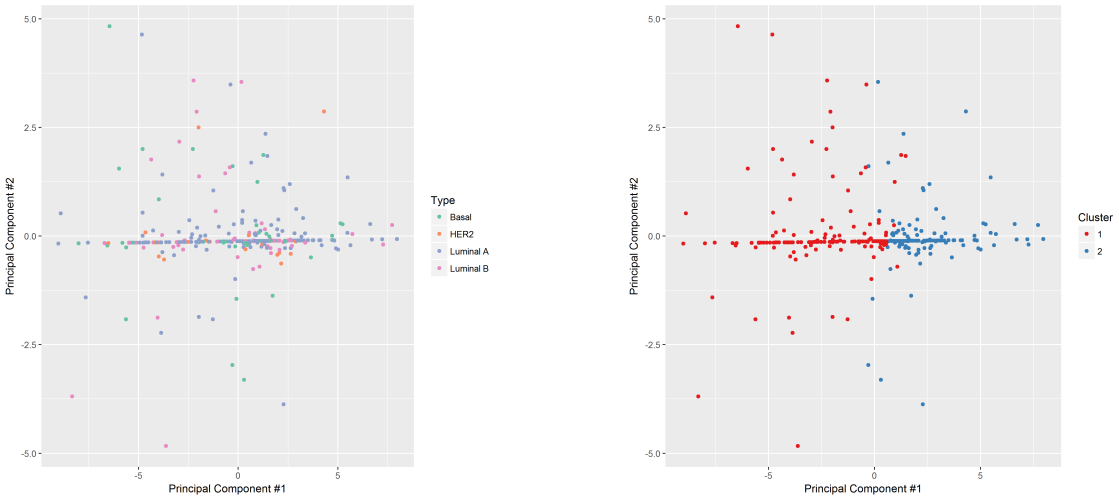
This chapter highlights the most relevant results obtained through the implementation and execution of the experimental design detailed in the previous chapter.

4.1 Initial Considerations

Prior to the analysis of any measured dependent variable, is worth mentioning some important results. Both the standalone miRNA Expression and Gene Copy Number datasets (coded as MiRNA and CN respectively) did not show any apparent clustering structure (with ARIs very close to 0). A possible explanation is that their features alone are not enough to make a good discrimination between groups. PCA plots show how unsuitable these datasets are for clustering. See figures 4.1 and 4.2. All algorithms performed poorly on this regard.

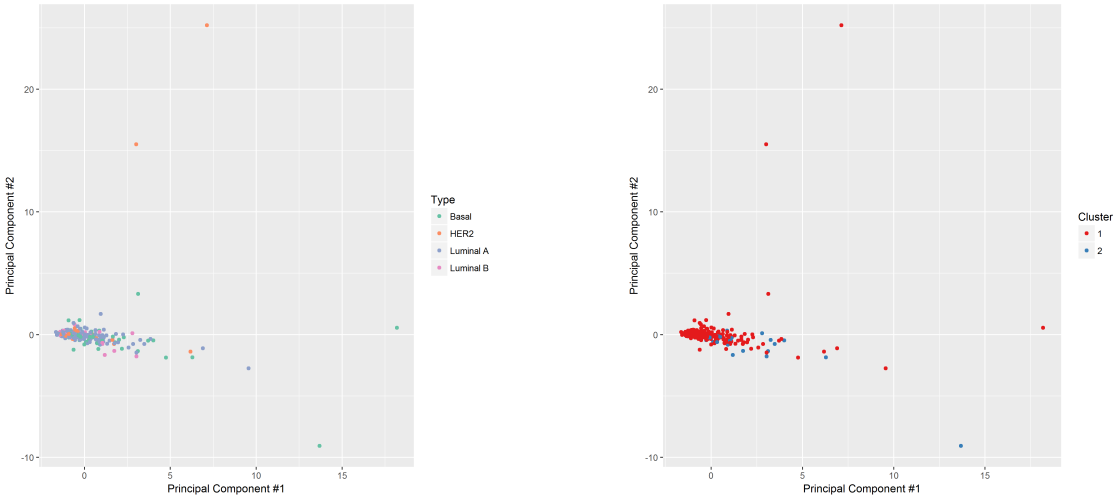
Even if clustering results seem bad, its worth noting that the data might serve for detecting outliers. Isolated points are clearly visible on both types of data.

The results associated with the standalone MiRNA and CN datasets are omitted in future calculations, as the average values of metrics could be highly distorted provoking a false sense of misquality.



(a) Groundtruth classification. (b) Hopfield classification.

Figure 4.1: Principal Component Analysis view of the sample from one of the experimental runs using the Gene Copy Number dataset with 10 features. Samples don't form any recognizable group.



(a) Groundtruth classification. (b) Hopfield classification.

Figure 4.2: Principal Component Analysis view of the sample from one of the experimental runs using the miRNA expression dataset with 10 features. All samples tend to cluster together into a single group.

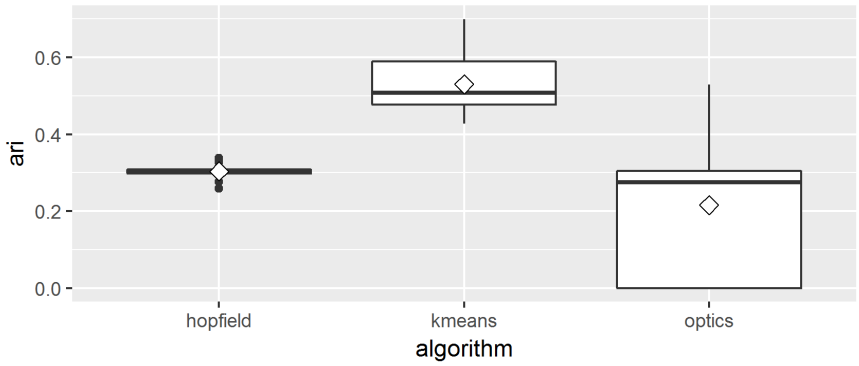


Figure 4.3: Box plot showing the distribution of ARI values for the evaluated algorithms. Means are marked with a diamond.

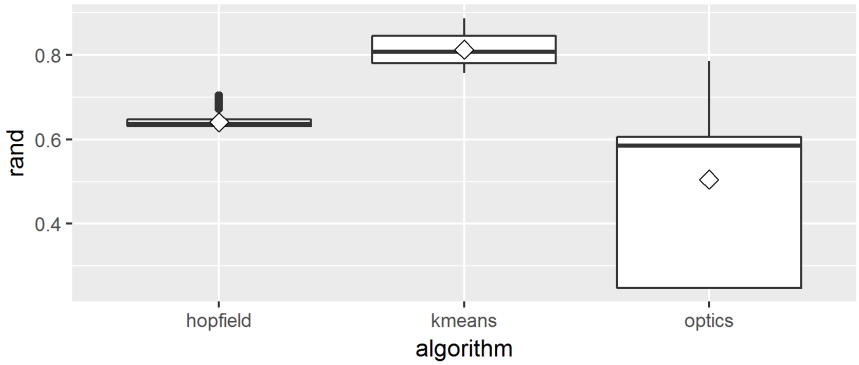


Figure 4.4: Box plot showing the distribution of RAND values for the evaluated algorithms. Means are marked with a diamond.

4.2 General Results

4.2.1 Clustering Accuracy

ARI and RAND values distributions for each algorithm are showed in figures 4.3 and 4.4.

Hopfield shows a mean ARI around 0.31. The algorithm appears to be very stable across all conditions and shows little variability. OPTICS, on the other hand seems to be highly variable with values ranging from 0 to 0.52, having a mean ARI around 0.23. This variability is product of the number of used features and dataset (see sections below). Kmeans shows the highest scores with a mean of 0.53. This is expected as Kmeans is hinted to create four clusters, which gives it an extra edge above the other algorithms. Still is interesting to make comparisons.

The figure 4.5 shows in a more detailed manner, the distribution of ARI values for all the possible experimental conditions. Points to highlight include:

1. Hopfield performance shows little to no variation, with the possible exception of the GE+CN with 100 features combination that shows a little.

2. OPTICS performance appears to get worse as more features are added. All the runs with 100 features got an ARI of 0 which means it detected a single cluster. On the other hand, GE+CN with 10 features got the better results with a maximum ARI of 0.51 which is considerable good.

3. Kmeans achieves its greatest performance using GE+MiRNA+CN with 100 features. It shows little variation under this configuration and reaches a top ARI of 0.68 which is unusually high for this type of data.

These points are explored more carefully in the sections below.

4.2.2 Number of Detected Clusters

The distribution for the detected number of clusters for each algorithm is shown in figure 4.6. It can be seen that both Hopfield and OPTICS detect two clusters most of the time. There is a small minority of times where Hopfield detects more than two (three, four or five). As will be seen later, only the case of three clusters is relevant, as the other two (four and five) incorporate clusters with a single element. OPTICS has a considerable number of cases with a single cluster (case with ARI of 0), and some with three. Kmeans always ends with four clusters because it was configured to do so.

4.2.3 Execution Time

The computer used for execution had a processor Intel(R) Core(TM) i5-2450M CPU 2.50GHz with 4,00 GB of RAM. Wall clock times for algorithms are showed in figure 4.7. It is clear that the Hopfield algorithm is very expensive in comparison with the others. This poses a major drawback to consider, as adding features appears to affect the execution time of Hopfield in a non-linear way.

Hopfield's performance could be explained by the following two points:

1. First of all, the algorithmic complexity. The one from Hopfield's might be several orders above in comparison with Kmeans and OPTICS which have implementations as good to be in $\Theta(n)$ and $\Theta(n \log(n))$ respectively. Only the recall phase from Hopfield requires a cost equivalent to several matrix multiplications, each having a cubic complexity; and this is repeated for each sample.

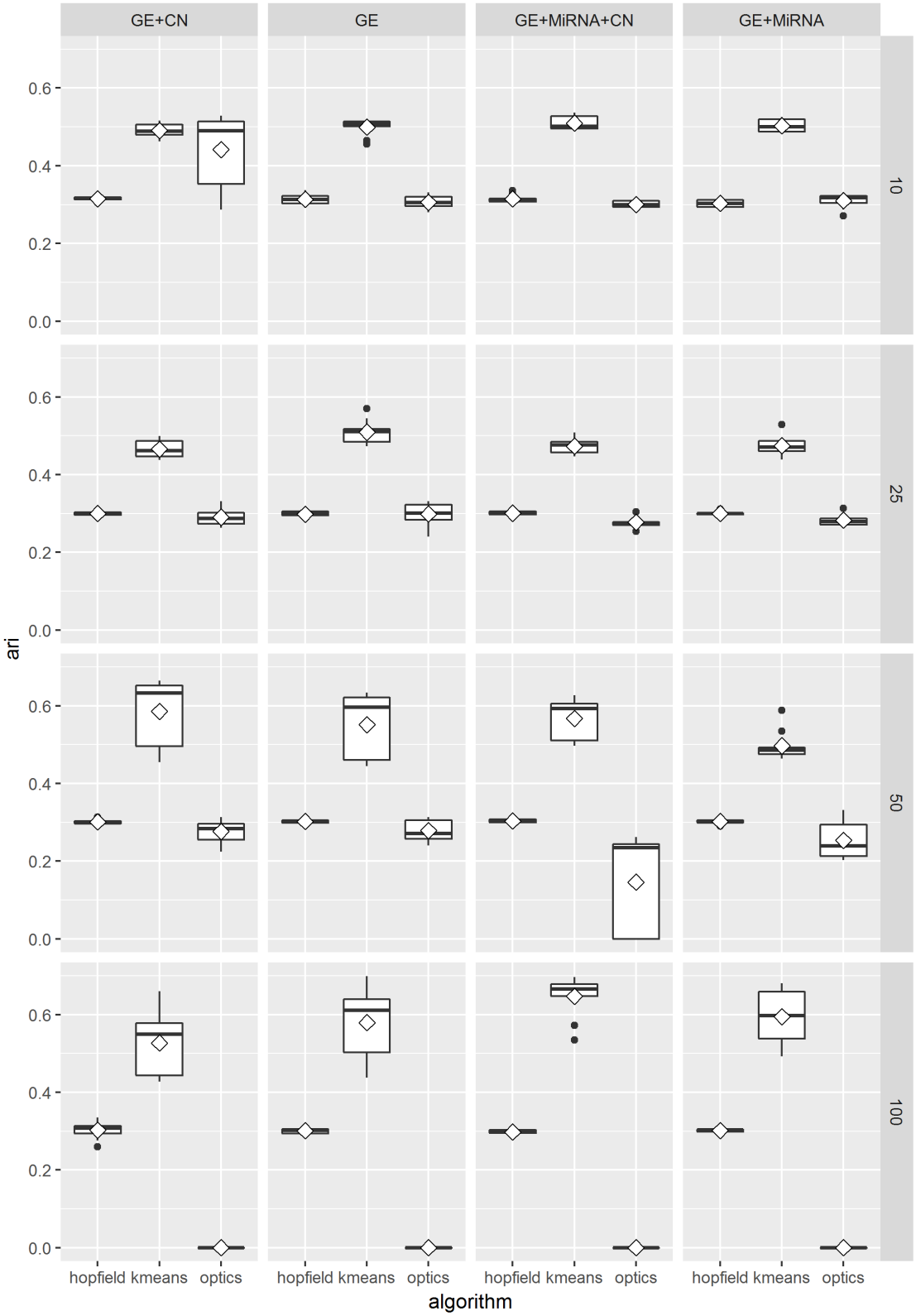


Figure 4.5: Box plot showing the distribution of ARI values for all the experimental conditions. X-axis varies the datasets, while y-axis the number of used features.

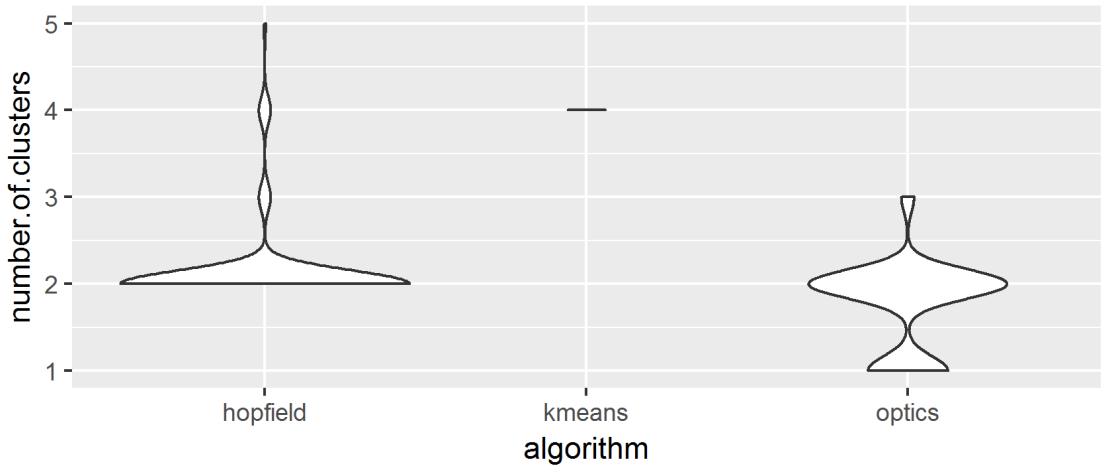


Figure 4.6: Violin plot showing the distribution for the detected number of clusters for each algorithm.

2. Hopfield's implementation was made completely in R and was no optimized by any means.

4.3 Discussion Points

4.3.1 Hopfield Remarks

Most of the experimental runs of Hopfield detect two clusters instead of the expected four. As a result, precision values are quite low (around 0.40). The recall values on the other hand are relatively high with a mean around 0.84. Inspecting some results visually, it seems that Hopfield clusters together all the luminal samples (both A and B types), and it does the same for the HER2 and basal ones. This can be easily appreciated in figure 4.8.

Inspecting the two generated attractors it can be seen that they are complementary (the sum of them gives a zero vector). This phenomenon appears to be a property of Hopfield Networks, that is, every attractor state has its polar opposite. However, this might not be desirable from a biological modeling point of view. The statement made in [22] of interpreting the attractor states as characteristic configurations of types of cancer might not be straightforward as it sounds. After all, it makes little sense to state that a type of cancer is a polar opposite of another in terms of gene expression.

There was a single experimental condition (GE + CN dataset with 100 features) where Hopfield obtained some conspicuous results. The emergence of a third cluster was clearly established as seen in figure 4.9. The third cluster appears to group a high number of luminal

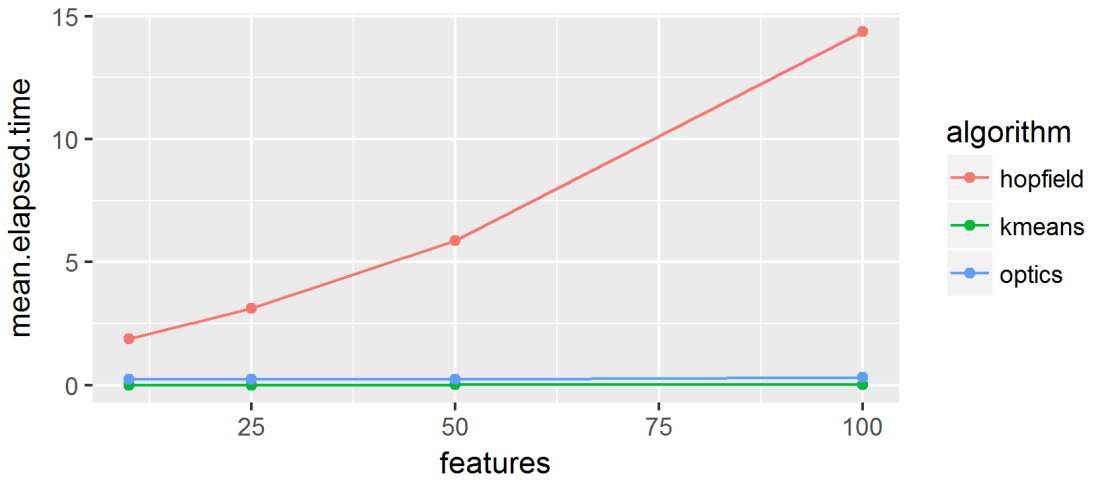
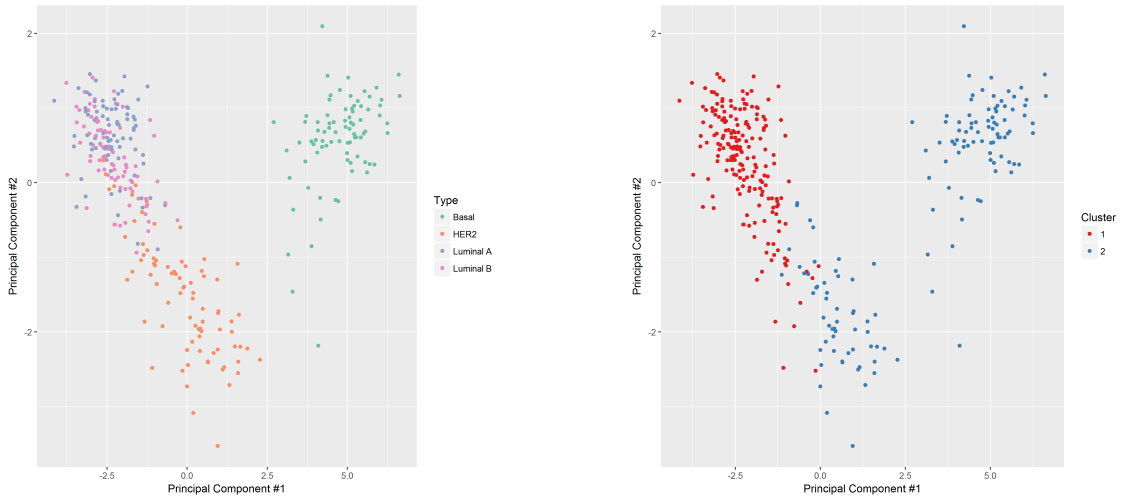


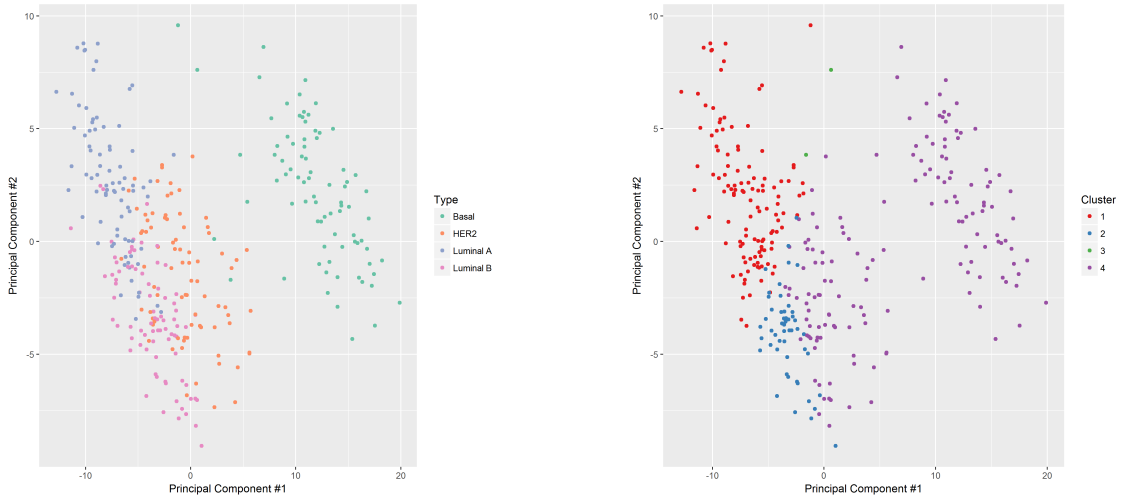
Figure 4.7: Lines plot showing the mean execution times for the evaluated algorithms. Time is measured in seconds.



(a) Groundtruth classification.

(b) Hopfield classification.

Figure 4.8: Principal Component Analysis view of the sample from one of the experimental runs using the Gene (mRNA) expression dataset with 10 features.



(a) Groundtruth classification. (b) Hopfield classification.

Figure 4.9: Principal Component Analysis view of the sample from one of the experimental runs using the GE + CN (gene expression values divided by copy number) dataset with 100 features.

B cases, which gave a slight improvement of precision and a considerable one in specificity (true negative rate increased from a mean value of 0.59 to 0.70). Still ARI values are similar to the other experimental conditions. Some single (or two) element clusters also appeared. Most probably, are cases of local minima in the Hopfield landscape. It would be interesting to validate if these cases correspond to outliers in some form.

Overall, the Hopfield Network's property of having at least two attractors might pose some doubts about its effectiveness for clustering. It might be argued, that the clustering structure is a result of the method and no of the data. Surprisingly, this might be useful in some cases for data that contains clusters connected by bridges (density based algorithms might fail on this regard). Still, a more careful evaluation is necessary.

Also, the fact that a third cluster appeared as the result of a high number of features (100) poses a interesting question in terms of how the number of features affects the quantity of attractors in the network. From an information point of view, a higher number of features means more capacity of having different states in the network [23]. In the context of biological problems that have a lot of variables, further exploration of this characteristic might be interesting.

4.3.2 OPTICS Remarks

Similar to the Hopfield algorithm, OPTICS only detected two clusters most of the time. However, while Hopfield tended to create a group exclusively for luminals and another for

(a) Groundtruth classification. (b) OPTICS classification.

Figure 4.10: Principal Component Analysis view of the sample from one of the experimental runs using the Gene (mRNA) expression dataset with 10 features.

HER2/basal samples, OPTICS separated the basal ones into their own cluster and put the rest of samples together. Notice that in figure 4.10 the achieved recall was perfect (equal to 1). This means that all element pairs that should go together are indeed grouped inside the same cluster. This is somewhat meritorious as long there is more than one cluster.

OPTICS in general, appears to behave worse as the number of features is increased. See figure 4.11. It even reaches the point of detecting a single cluster when using 100 features. From the PCA view (figure 4.12) it can be seen that points become less coupled together in a high dimensional space, so it can be understood that a density-based approach for clustering might fail.

The experimental condition that obtained the best results for OPTICS (and most interesting, overall) was using the GE + CN dataset with only 10 features. Under this configuration it detected three well-defined clusters on seven out of the ten runs. It managed to group the majority of HER2 cases into a separate group (see figure 4.13).

Ten is a remarkable low number of features for a cancer diagnostic test (see Table 4.1 for the gene list). For example PAM50 uses 50, and Oncotype DX uses a 21 gene-assay. Even if the model does not differentiate between luminal A and B, the low number of required features somehow compensates. These results, mixed with the fact that were only obtainable by dividing gene expression values by the copy number ones, look promising for further exploration and research. It hints at the possibility of implementing good diagnostic tests at a lower cost.

In the PCA biplot of the figure 4.14 the correlations between variables and how they impact the different groups can be appreciated. The basal group is strongly affected by the expres-

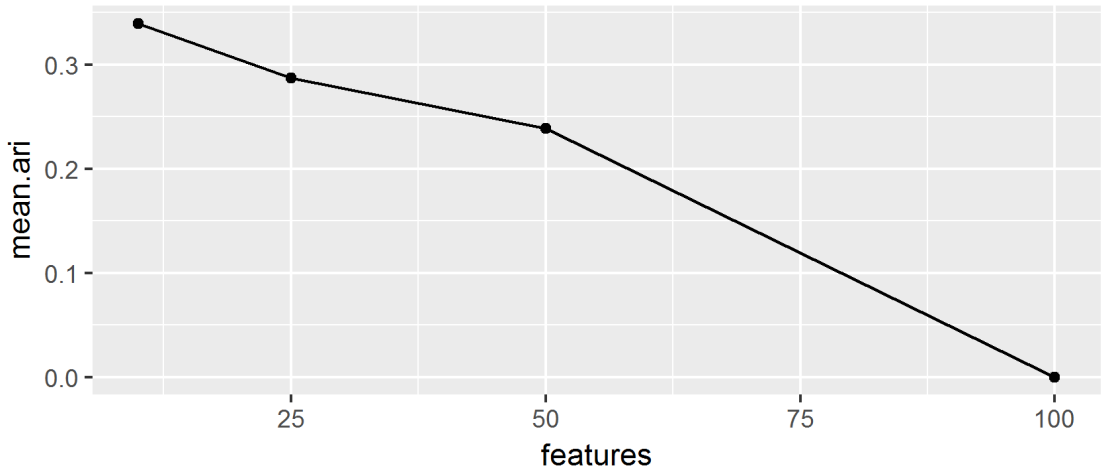
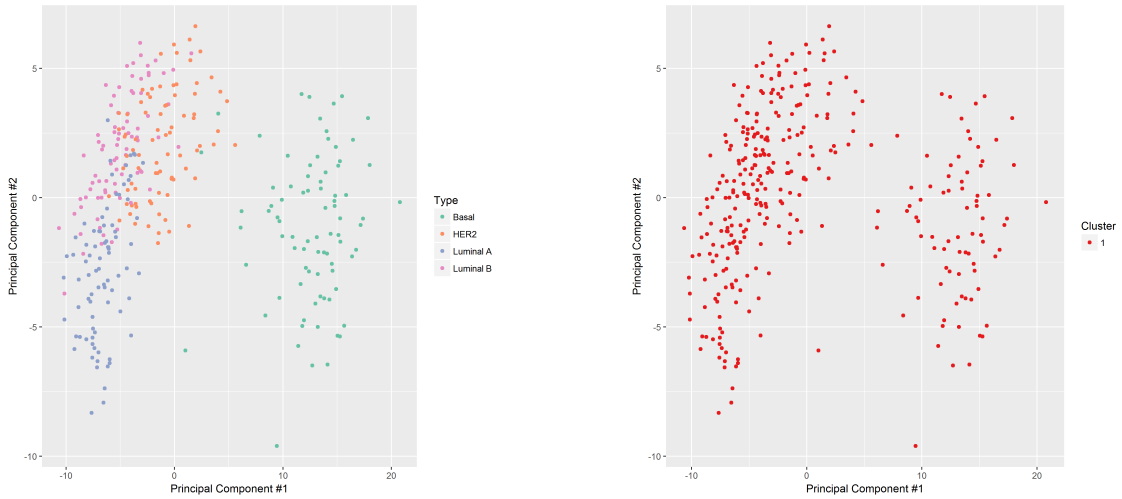
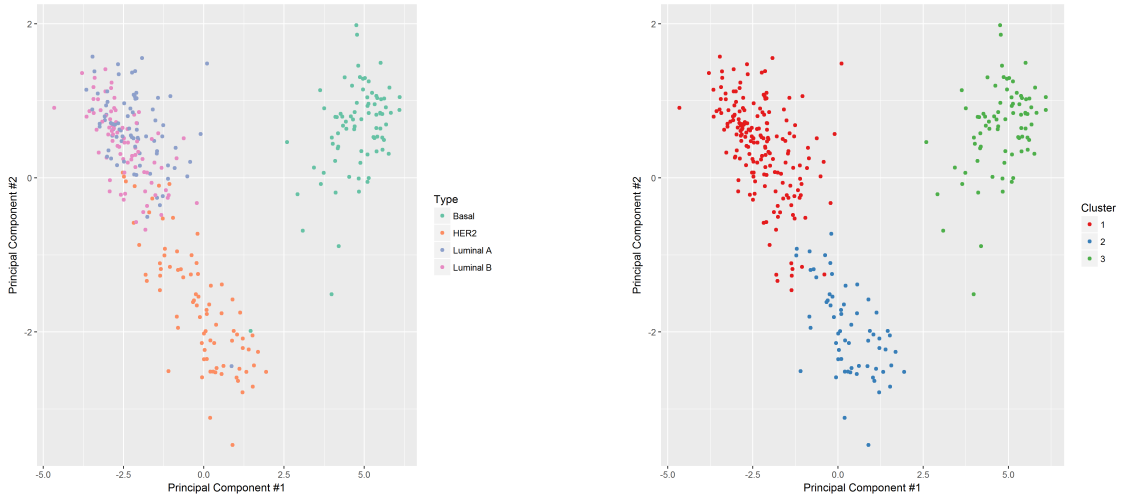


Figure 4.11: Line plot showing the mean ARI values for OPTICS algorithm while varying the number of used features.



(a) Groundtruth classification. (b) OPTICS classification.

Figure 4.12: Principal Component Analysis view of the sample from one of the experimental runs using the Gene (mRNA) expression dataset with 100 features.



(a) Groundtruth classification.

(b) OPTICS classification.

Figure 4.13: Principal Component Analysis view of the sample from one of the experimental runs using the GE + CN (gene expression values divided by copy number) dataset with 10 features.

Gene ID	Gene Name	Gene Synonyms	Description
ENSG00000054598	FOXC1	ARA, FKHL7, FREAC3, IGDA, IHG1, IRID1	forkhead box C1
ENSG00000074410	CA12	HsT18816	carbonic anhydrase 12
ENSG00000091831	ESR1	ER-alpha, ESR, Era, NR3A1	estrogen receptor 1
ENSG00000107485	GATA3	HDR	GATA binding protein 3
ENSG00000109436	TBC1D9	GRAMD9, KIAA0882, MDR1	TBC1 domain family member 9
ENSG00000115648	MLPH	Slac-2a, exophilin-3, l(1)-3Rk, l1Rk3, ln	melanophilin
ENSG00000124664	SPDEF	PDEF, bA375E1.3	SAM pointed domain containing ETS transcription factor
ENSG00000129514	FOXA1	HNF3A	forkhead box A1
ENSG00000173467	AGR3	BCMP11, HAG3, PDIA18, hAG-3	anterior gradient 3
ENSG00000182175	RGMA	RGM, RGMa	repulsive guidance molecule BMP co-receptor a

Table 4.1: The ten genes used for OPTICS best performance scenario. Information taken from “ensembl.org” website.

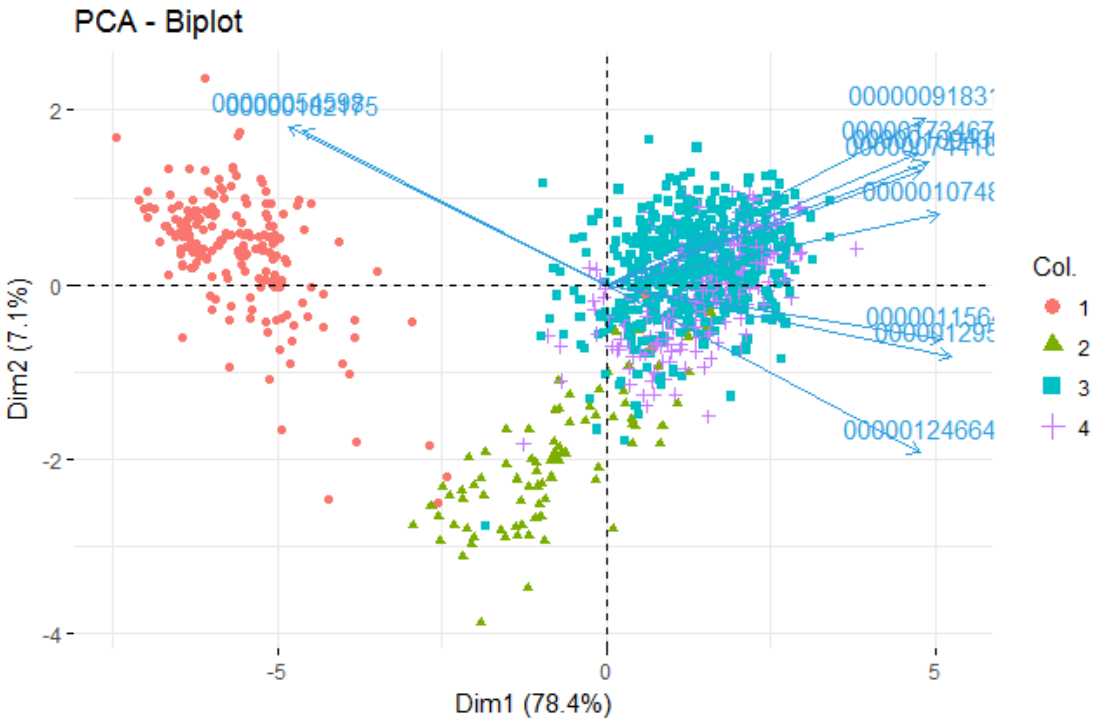
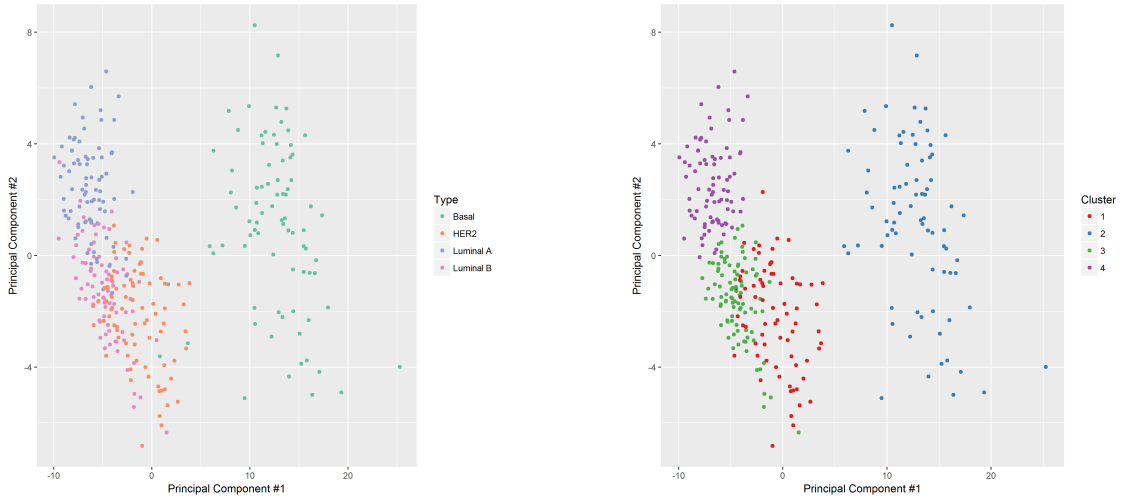


Figure 4.14: A PCA biplot using the majority of breast cancer samples from the TCGA with the 10 genes GE + CN dataset. Red points are basal samples, light green triangles are HER2, squares and crosses are luminal.



(a) Groundtruth classification. (b) Kmeans classification.

Figure 4.15: Principal Component Analysis view of the sample from one of the experimental runs using the GE+MiRNA+CN dataset (gene expression values divided by copy number and also integrated with miRNA variables) with 100 features.

sion of FOXC1 and RGMA genes, and posses low expression rates for MLPH and FOXA1. Luminal samples are strongly affected by the expression of several genes, most notoriously the ESR1. There are no arrows pointing towards the HER2 group, so these are discriminated by some of the genes highly expressed in the luminal samples, but not in them. The biplot also reveals arrows that are very close together (features with high positive correlation). A strong correlation between variables reveals a certain degree of redundancy between them. All these details might be used to improve the selected features.

4.3.3 Kmeans Remarks

Kmeans, maybe in an expected way, obtained the highest ARI values. The precision was improved merely by the fact of hinting the algorithm to produce four groups (the same number of clusters in the groundtruth). Despite this, the algorithm showed great variability across runs in the way it clustered the samples. On some runs it divided the basal samples into two separate groups, and on others the luminal ones (not reflecting the A and B subtypes). Most of the times, a representative cluster for the HER2 type was created.

Curiously, the best results for Kmeans were show using the GE+MiRNA+CN dataset (that is, the three types integrated together) with 100 features. One of the runs reached an ARI of 0.68 and a precision of 0.91, the highest among all the runs for all the configurations (see figure 4.15). Under this configuration the variation phenomena described in the paragraph above it's still present across runs, with the difference that the group of luminal samples, if divided, it is better separated into the A and B subtypes.

How much the miRNA variables really contributed to the results is something not entirely clear at first glance. Given the little discriminative power that miRNA variables showed (via their BSS/WSS ratios) in comparison to the mRNA ones, the result is kind of interesting. A possibility is that miRNA variables did introduce less noise (hence the unexpected improvements), on which case the contribution to the separation of classes might not be significant.

Chapter 5

Conclusions and Future Work

This chapter concludes with the most important remarks and lessons acquired through the elaboration of this research.

5.1 General Conclusions

Cancer is still one of the greatest challenges humankind has ever faced. However, if we look in retrospective, the understanding that has been generated in recent years is considerable and trans-disciplinary action has played a key role on it. Molecular biologists cannot cope with the amounts of data that are being generated and need the aid of computational tools both for processing and analyzing patterns. Clustering is only one example of how the "divide and conquer" approach is useful for tackling something as complex as cancer. After all, breaking a disease down into different scenarios helps to structure strategies for treatment.

Public repositories of data, such GDC/TCGA, have opened the opportunities for new type of studies to be made. The quantity and quality of data are factors to consider when trying to answer a biological question. Fortunately, having a common repository available for the scientific community helps in terms of reproducibility and standardization. Of course, having a great diversity of omic data types (mRNA, miRNA, copy number variation...) introduces the challenge of figuring out how to use them together to gain a more solid understanding of the dynamics underneath. There is still room for trying new data integration strategies, such as in this study, where dividing gene expression values by their corresponded copy number improved the discrimination of breast cancer types in some scenarios. This in turn poses new research questions. Could it be that normalizing the gene expression in that way minimizes the noise caused by the over expression of some genes?

Regarding the topic of clustering, definitely there is no best universal algorithm. It entirely

depends of several factors like: the shape of data, the quantity of noise and the external knowledge about the problem. Gene expression is a particularly difficult type of data to dealt with, beginning by the fact that it has an overwhelming number of features that do not follow uniform distributions by any means. Normalization and feature selection are vital processes for a successful data analysis in genomics, and are as important, if not more, than the training/classification algorithm itself.

The Hopfield clustering algorithm studied on this research had certain characteristics that made it appealing for experimenting on genomic data. The most notorious of these characteristics was its convergence dynamics which might resemble a cellular differentiation process. However, the mirror attractor phenomena present in the network (every stable state has its polar opposite) puts the biological modeling in question, as it makes no sense on a real scenario. Even if samples are perfectly clustered, the information contained in the attractor states might be misleading, or at least, should be interpreted very carefully. Also, its remarkable strange having a clustering algorithm that has no parameter other than the data, which is binarized and subject to information loss. On the bright side, the algorithm has interesting properties, and appears to have more information storage capabilities as more features are added (states have more coding power). This characteristic, plus the fact that the hebbian learning scheme does not depend on a distance measure, might render the algorithm useful for problems with a high number of features. Still, the complexity of the algorithm poses a drawback to consider if implemented.

Finally, despite all the objectives of the study being fulfilled, the main hypothesis did not hold completely:

> Through the use of multi-omic data from the TCGA and a hebbian learning scheme, is possible to create a Hopfield Network model for breast cancer subtyping and characterization. This model will either differentiate between the main four molecular subtypes of breast cancer, or will find a finer classification inside these.

A model for breast subcancer was created indeed, but was not able to differentiate completely between the main four molecular subtypes, and even less, a subclassification inside them. In addition, the "characterization" aspect of the model is affected by the properties of the attractors and their opposites. However, valuable new hypothesis can be derived, that is, the possibility of creating a model with few genes that discriminates between luminal (both A and B), basal and HER2 groups.

5.2 Limitations

There were two main limitations for the study:

1. First, a simple and single strategy was used for feature selection. There are a good variety of algorithms in the literature. At least other two can be used and contrasted.

2. Second, the miRNA integration strategy did not suit well the feature selection method that was used. This is because the BSS/WSS ratio of miRNA variables is relatively low if compared to the mRNA ones. If a single ranking of variables is applied then the miRNA features are left out completely. This fact hints that there could be more intelligent ways of integrating this type of data.

5.3 Future Work

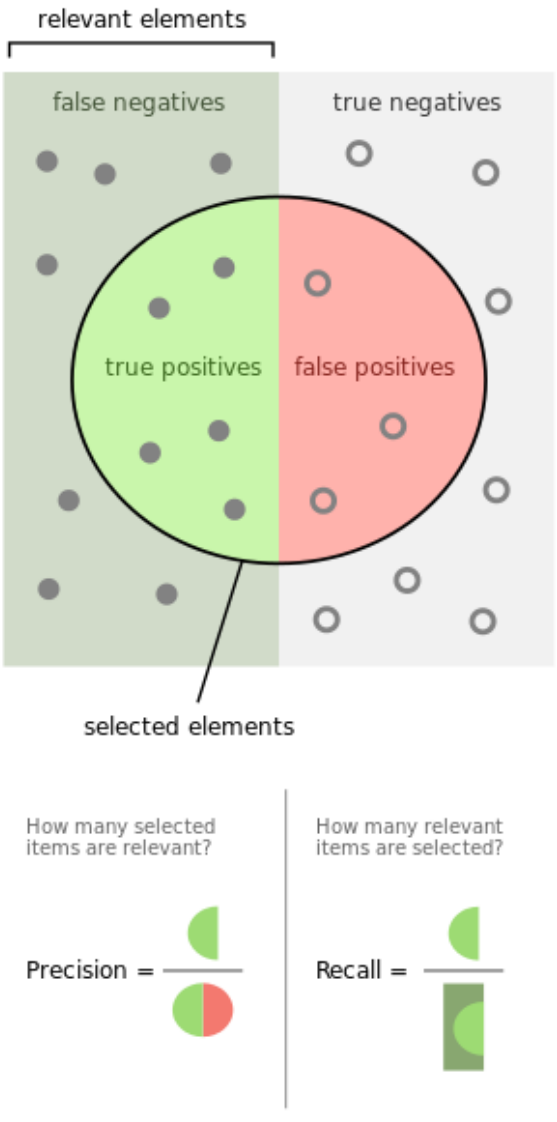
The most relevant result, maybe in an ironical way, was obtained not by using the Hopfield Network, but the OPTICS algorithm. This result, hinted at the possibility of having a precise prediction model with a low quantity of features (10 on this case). Even if the model does not differentiate between luminal A and B subtypes, the good discrimination it makes for other types and its low cost makes it promising for further exploration. In a very optimistic scenario, a diagnostic test that uses only 10 genes would be very economical in comparison to other tests like PAM50 which use 50.

There are several ways the model could be improved. A close examination of the optimal number of features should be made, this time putting emphasizing in a narrow range like $5 \ldots 15$. Knowing that is possible to separate the samples into three well-formed groups (luminal, HER2 and basal), supervised models could also be explored (using a single "luminal" label instead of two). Also, in order to relax the problem of differentiating between luminal A and B, a separate model could made with the exclusive task of discriminating between these two groups. That would be used in a second step if the first model dictates a "luminal" output. As long the total number of used genes is relatively low, then having a hierarchical organization of predictive models is a viable option.

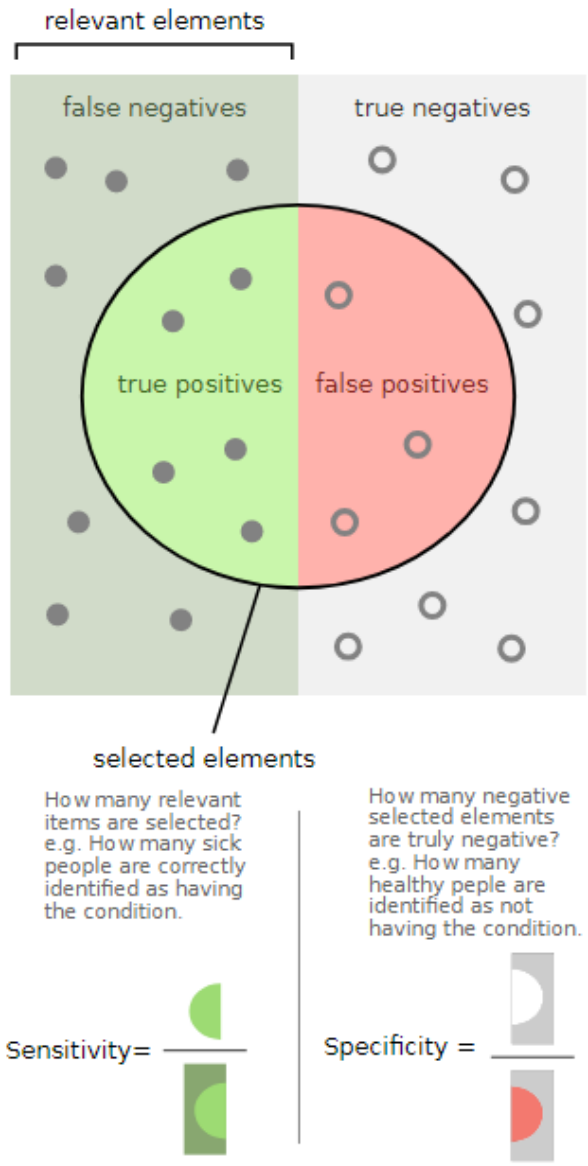
Chapter 6

Annexes

6.1 Precision and Recall



6.2 Sensitivity and Specificity



Bibliography

[1] a a Alizadeh, M B Eisen, R E Davis, C Ma, I S Lossos, A Rosenwald, J C Boldrick, H Sabet, T Tran, X Yu, J I Powell, L Yang, G E Marti, T Moore, J Hudson, L Lu, D B Lewis, R Tibshirani, G Sherlock, W C Chan, T C Greiner, D D Weisenburger, J O Armitage, R Warnke, R Levy, W Wilson, M R Grever, J C Byrd, D Botstein, P O Brown, and L M Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–11, 2000.

[2] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod Record*, pages 49–60, 1999.

[3] Jordi Barretina, Giordano Caponigro, and Nicolas Stransky. The Cancer Cell Line Encyclopedia enables predictive modeling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, 2012.

[4] Dennis A. Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J. Lipman, James Ostell, and Eric W. Sayers. GenBank. *Nucleic Acids Research*, 41(D1):36–42, 2013.

[5] Kyle Chang, Chad J Creighton, Caleb Davis, Lawrence Donehower, et al. The Cancer Genome Atlas Pan-Cancer analysis project. *Nature Genetics*, 45(10):1113–1120, 2013.

[6] Mike Cusnir and Ludmila Cavalcante. Inter-tumor heterogeneity. (August):1143–1145, 2012.

[7] Xiaofeng Dai, Ting Li, Zhonghu Bai, Yankun Yang, Xiuxia Liu, Jinling Zhan, and Bozhi Shi. Breast cancer intrinsic subtype classification, clinical use and future trends. *American Journal of Cancer Research*, 5(10):2929–2943, 2015.

[8] Ministerio de Salud. *Plan Nacional para la Prevención y Control del Cáncer 2011-2017*. 2012.

[9] Marcilio C P de Souto, Ivan G Costa, Daniel S A de Araujo, Teresa B Ludermir, and Alexander Schliep. Clustering cancer gene expression data: a comparative study. *BMC Bioinformatics*, 9(1):497, 2008.

[10] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[11] R. Edgar. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002.

[12] Atefeh Taherian Fard and Mark A. Ragan. Modeling the attractor landscape of disease progression: A network-based approach. *Frontiers in Genetics*, 8(APR):1–11, 2017.

[13] Atefeh Taherian Fard, Sriganesh Srihari, Jessica C Mar, and Mark A Ragan. Not just a colourful metaphor: modelling the landscape of cellular development using Hopfield networks. *npj Systems Biology and Applications*, 2(November 2015):16001, 2016.

[14] Christina Fitzmaurice, Christine Allen, Ryan M. Barber, Lars Barregard, Zulfiqar A. Bhutta, et al. Global, Regional, and National Cancer Incidence, Mortality, Years of Life Lost, Years Lived With Disability, and Disability-Adjusted Life-years for 32 Cancer Groups, 1990 to 2015. *JAMA Oncology*, 3(4):524, 2017.

[15] T. R. Golub. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286(5439):531–537, 1999.

[16] Douglas Hanahan and Robert A Weinberg. The Hallmarks of Cancer Review University of California at San Francisco. 100:57–70, 2000.

[17] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[18] J Larry Jameson. Oncogenes and tumor suppressor genes. In *Principles of molecular medicine*, pages 73–82. Springer, 1998.

[19] G. Kerr, H.J. Ruskin, M. Crane, and P. Doolan. Techniques for clustering gene expression data. *Computers in Biology and Medicine*, 38(3):283–293, 2008.

[20] Susan G Komen. Molecular subtypes of breast cancer. `http://ww5.komen.org/BreastCancer/SubtypesofBreastCancer.html`.

[21] Alexei A. Koulakov and Yuri Lazebnik. The problem of colliding networks and its relation to cell fusion and cancer. *Biophysical Journal*, 103(9):2011–2020, 2012.

[22] Stefan R. Maetschke and Mark A. Ragan. Characterizing cancer subtypes as attractors of Hopfield networks. *Bioinformatics*, 30(9):1273–1279, 2014.

[23] Robert J. Mceliece, Edward C. Posner, Eugene R. Rodemich, and Santosh S. Venkatesh. The Capacity of the Hopfield Associative Memory. *IEEE Transactions on Information Theory*, 33(4):461–482, 1987.

[24] Andrew McGuire, James Brown, Carmel Malone, Ray McLaughlin, and Michael Kerin. Effects of Age on the Detection and Management of Breast Cancer. *Cancers*, 7(2):908–929, 2015.

[25] Marina Meil?? Comparing clusterings-an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[26] Boris Mirkin. *Clustering: a data recovery approach*. CRC Press, 2012.

[27] NIH. Breast cancer. `https://www.cancer.gov/types/breast`.

[28] NIH. Genomic data commons. `https://gdc.cancer.gov/`.

[29] NIH. What is cancer? `https://www.cancer.gov/about-cancer/understanding/what-is-cancer`.

[30] NIH. Nih working definition of bioinformatics and computational biology. `https://www.kennedykrieger.org/sites/default/files/research_related_files/bioinformatics-def.pdf`, 2000.

[31] Diane M. Pereira, Pedro M. Rodrigues, Pedro M. Borralho, and Cecília M P Rodrigues. Delivering the promise of miRNA cancer therapeutics. *Drug Discovery Today*, 18(5-6):282–289, 2013.

[32] Mehdi Pirooznia, Jack Y Yang, Mary Qu Yang, and Youping Deng. A comparative study of different machine learning methods on microarray gene expression data. *BMC Genomics*, 9(Suppl 1):S13, 2008.

[33] Sai Teja Pusuluri. Exploring Neural Network Models with Hierarchical Memories and Their Use in Modeling Biological Systems. (April), 2017.

[34] Sai Teja Pusuluri, Alex Hunter Lang, Pankaj Mehta, and Horacio Emilio Castillo. Controlling energy landscapes with correlations between minima. *arXiv*, page 1611.06127, 2016.

[35] Davide Risso, Katja Schwartz, Gavin Sherlock, and Sandrine Dudoit. GC-Content Normalization for RNA-Seq Data. *BMC Bioinformatics*, 12(1):480, 2011.

[36] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[37] Anna V Roschke and Ilan R Kirsch. Targeting karyotypic complexity and chromosomal instability of cancer cells. *Current drug targets*, 11(10):1341–50, 2010.

[38] Angela Serra, Michele Fratello, Vittorio Fortino, Giancarlo Raiconi, Roberto Tagliaferri, and Dario Greco. MVDA: a multi-view genomic data integration methodology. *BMC Bioinformatics*, 16(1):261, 2015.

[39] T. Sorlie, C. M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, T. Thorsen, H. Quist, J. C. Matese, P. O. Brown, D. Botstein, P. E. Lonning, and A.-L. Borresen-Dale. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, 98(19):10869–10874, 2001.

[40] Christos Sotiriou, Soek-Ying Y Neo, Lisa M McShane, Edward L Korn, Philip M Long, Amir Jazaeri, Philippe Martiat, Steve B Fox, Adrian L Harris, and Edison T Liu. Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proceedings of the National Academy of Sciences of the United States of America*, 100(18):10393–10398, 2003.

[41] Anthony Szedlak, Giovanni Paternostro, and Carlo Piermarocchi. Control of asymmetric hopfield networks and application to cancer attractors. *PLoS ONE*, 9(8), 2014.

[42] Phillippa C. Taberlay and Peter A. Jones. DNA methylation and cancer. *Progress in Drug Research*, 67(22):1–23, 2011.

[43] Erdogan Taskesen, Sjoerd M H Huisman, Ahmed Mahfouz, Jesse H Krijthe, Jeroen De Ridder, Anja Van De Stolpe, Erik Van Den Akker, Wim Verheagh, and Marcel J T Reinders. Pan-cancer subtyping in a 2D- map shows substructures that are driven by specific combinations of molecular characteristics. *Nature Publishing Group*, 2016.

[44] The Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumors. *Nature*, 490(7418):61–70, 2012.

[45] Cole Trapnell, Brian A. Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J. Van Baren, Steven L. Salzberg, Barbara J. Wold, and Lior Pachter. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5):511–515, 2010.

[46] Laura J. van 't Veer, Hongyue Dai, Marc J. van de Vijver, Yudong D. He, Augustinus A. M. Hart, Mao Mao, Hans L. Peterse, Karin van der Kooy, Matthew J. Marton, Anke T. Witteveen, George J. Schreiber, Ron M. Kerkhoven, Chris Roberts, Peter S. Linsley, René Bernards, and Stephen H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, 2002.

[47] Chenwei Wang, Alperen Taciroglu, Stefan R Maetschke, Colleen C Nelson, Mark A Ragan, and Melissa J Davis. mCOPA: analysis of heterogeneous features in cancer expression data. *Journal of clinical bioinformatics*, 2(1):22, 2012.

[48] Hadley Wickham. Data Tidying. 11:1–23, 2016.

[49] Ruoshi Yuan, Xiaomei Zhu, Gaowei Wang, Site Li, and Ping Ao. Cancer as robust intrinsic state shaped by evolution: a key issues review. *Reports on Progress in Physics*, 80(4):042701, 2017.

[50] Xi Zhao, Einar Andreas Rødland, Robert Tibshirani, and Sylvia Plevritis. Molecular subtyping for clinically defined breast cancer subgroups. *Breast Cancer Research*, 17(1):29, 2015.