Unsupervised Clustering for Mining Patterns in Genomic Dataset

Master Thesis
for the award of the degree of
Master in Artificial Intelligence

FACULTAT D’INFORMÀTICA DE BARCELONA (FIB)
FACULTAT DE MATEMÀTIQUES (UB)
ESCOLA TÈCNICA SUPERIOR D’ENGINYERIA (URV)

Gonzalo Gómez Sánchez
(gonzalo.gomez@bsc.es)

Advisor
Josep Ll. Berral García
(josep.berral@bsc.es)

Co-advisor
David Carrera Pérez
(david.carrera@bsc.es)

Data-Centric and Computing Department (DCC)

April 13, 2018
Abstract

The human genome hides all the information about the functions of our body. However, getting to know the meaning of all its sequences is far from today. The good thing is that we do not need to know all the meaning of the genome to know that some chromosomes or some sections may offer particular behaviours that could indicate suffering a particular illness or having a specific malfunctioning cell. These phenomena are known as bio-markers and finding them is a big point of interest for the development of personalized medicine. The aim of this thesis is to develop a method to automatically find a specific bio-marker in a genomic dataset of translocations. This specific bio-marker has been proposed by the Life Science (LS) department of the Barcelona Supercomputing Center: a chain of chromosome rearrangements following the shape of a triangle, where 3 different rupture points of the chromosome share all their genetic information. The proposed architecture for detecting this kind of patterns uses an unsupervised clustering method based on Kernel Density Estimation functions and Graph Mining techniques for the detection of the patterns, both of them developed using distributed computing techniques. Finally, the architecture has been tested with a genomic dataset, proving its well-function by the domain experts (LS researches) and detecting a potential bio-marker for sarcoma in chromosome 12 after doing an exploratory analysis of the data provided by the system.
Acknowledgments

I would like to thank my thesis advisor Josep Ll., for being always supportive and having all the answers to my doubts. To my co-advisor David, for giving me the opportunity of working in a project I believe in. I would also like to thank ML group for sharing our concerns in the meetings and helping each others. Finally, to LS researches, for the trust deposited on our work.
# Contents

List of Figures ii
List of Tables iii

1 Introduction 1
  1.1 Motivation .............................................. 3
  1.2 Objectives .............................................. 4
  1.3 Contributions .......................................... 5
  1.4 Document organization ................................... 5

2 Background 6
  2.1 Dataset ................................................. 6
  2.2 Joining breakpoints ....................................... 8
    2.2.1 Life Science approach ............................... 8
    2.2.2 Starting point ...................................... 10

3 Related work 11

4 Clustering Architecture 14
  4.1 Data preparation ......................................... 14
  4.2 KDE distributed clustering ................................ 15
  4.3 Solving lack of resolution: re-clustering ................. 18
    4.3.1 Cluster cleaning .................................... 18
    4.3.2 Re-clustering ....................................... 18
  4.4 Final Rearrangements .................................... 20

5 Exploratory Analysis 23
  5.1 Finding triangles ........................................ 23
    5.1.1 Graph Mining ....................................... 23
  5.2 Exploratory Analysis ..................................... 27
    5.2.1 Triangle analysis per Cancer type .................. 28
    5.2.2 Triangle locations .................................. 28

6 Discussion 37

References 40
List of Figures

1.1 Human Karyotype ........................................... 1
1.2 Possible chromosome rearrangements [1] .................. 2
1.3 Approximation of the cost per genome through past years ... 3
1.4 Triangular rearrangement ................................... 4
1.5 Block diagram of the proposed architecture ............... 5

2.1 Chromosome size in base-pairs ............................. 7
2.2 Dataset sample example .................................... 7
2.3 Inter-window and intra-window concepts ................... 9

4.1 Kernel Density Estimation example ........................ 16
4.2 Kernel Density Estimation example ........................ 17
4.3 Re-clustering process. In the top of the figure we can see how a KDE clustering is performed inside every cluster (after the cleaning process), with the aim of obtaining smaller clusters that fits better to the joining problem. In the bottom, it can be seen the final clusters of the chromosome after the re-clustering. ............................................ 19
4.4 Final clustering process. Edges refer to chromosomal rearrangements between pairs of breakpoints. ............... 21

5.1 Example graph and the different steps of the sub-graph mining algorithm for one of the vertices. This process is repeated for the four of them. ................................. 26
5.2 Blocks diagram of the graph mining algorithm. .......... 27
5.3 Histogram of triangle breakpoints locations vs chromosome size. Both values have been normalized. .................... 30
5.4 Cancer types ranked by anomalies in their samples ....... 32
5.5 Histogram of triangle breakpoints locations vs chromosome size per cancer type. Both values have been normalized. .......... 34
5.6 Histogram of triangle breakpoints locations vs chromosome size per cancer type. Sorted by degree of anomaly. Both values have been normalized. ......................... 35
5.7 Histogram of triangle breakpoints locations vs chromosome size for SARC samples. ................................. 36
List of Tables

4.1 Clustering sample table ............................................. 15
4.2 Post-KDE clustering sample table example ......................... 17
4.3 Percentils of intra-cluster breakpoints distance ................... 18
4.4 Re-clustering bandwidths comparison. For every bandwidths
   has been obtained the number of samples with triangles ob-
   tained, the number of samples with triangles obtained with
   both LS and the proposed method and the distance between
   breakpoints inside clusters at the 95 percentil. Final band-
   width decision has been also based on all the percentil val-
   ues but only the most relevant for this case (95) is shown. .... 20

5.1 Sample rearrangements clustered before (left) and after (right)
   eliminating duplicates. ............................................. 24
5.2 The found sub-graphs are saved along with their canoni-
   cal labels and edges involved. If a triangle is found twice,
   it should have both same canonical label and same edge’s
   hash. It will be removed from the table (bold rows). ............ 25
5.3 Statistic analysis of triangle frequency over cancer type. C.
   type corresponds with cancer type and s.w.tr. to samples
   with triangles. ...................................................... 29
5.4 Stats of cancer types with high anomalies. Anomalous chro-
   mosome, number of samples of cancer type, number of sam-
   ples with triangles in anomalous chromosome and number
   of samples with maximum number of triangles in anom-
  alous chromosome .................................................. 33
Chapter 1

Introduction

Genomics is the field of study of the genome. The word genome refers to all the DNA contained in the cells of our body. The DNA or deoxyribonucleic acid contains all the genetic information needed for the build, work, and repair of all living organism. This information is stored as a code made up of four chemical bases. Human DNA consists of 3 billion bases and are almost equal to every human. All this information is stored in the nucleus of each cell, packaged into thread-like structures called chromosomes. The complete set of human chromosomes is called Human Karyotype and is composed by 23 chromosomes fig. 1.1. Each one is made up of DNA tightly coiled many times around proteins called histones that support its structure. They have a constriction point called the centromere, which divides the chromosome into two sections called ‘arms’.

This chromosomes composed by genetic code can suffer rearrangements. Rearrangement is a breakage of DNA double helices in the genome at two different locations, followed by a rejoining of the broken ends to produce a new chromosomal arrangement of genes, different from the gene order of the chromosomes before they were broken [2]. This events
are classified as: deletion, duplication, inversion and translocation fig. 1.2. The interest on learning as much as possible about the chromosome rearrangements is mostly due to their direct contribution to cancer development. "Cancer" is the name given to a collection of related diseases where some of the body’s cells begin to divide without stopping and spread into surrounding tissues. This cell behaviour is due to DNA alterations, that are caused by the chromosome rearrangements.

![Figure 1.2: Possible chromosome rearrangements [1].](image)

The amount of data being produced by sequencing, mapping and analyzing genomes propels Genomics into the realm of Big Data. The volume of data produced by Genomics is huge; each human genome amounts to 100 gigabytes of data. Furthermore, since the first human genome took 13 years and around $500 million to be sequenced, the technology has progressed quite a bit and now it takes around 15 minutes and $1000 1.3 thanks to the Next Generation Sequencing (NGS) technologies [3]. Therefore, the amount of data available grows every day. However, this data has been sequenced but not decoded. This means that we have the information about which base-pairs go at every place but we do not know the meaning of this information. Decoding the meaning of all the parts of the genome is something far from today’s knowledge. The information that we have and can be useful is about the differences between the genomes, due to the rearrangements mentioned before. Specific rearrangements happening consistently in different samples suffering the same malfunction are considered as bio-markers of that malfunction. Knowing these bio-markers can help to detect this malfunctions or even decide which kind of treatment is more appropriate for different cases of the same malfunction.

Large-Scale Genomic Datasets contains these kind of information. However, it is difficult to obtain answers to research questions of this volume
of data by the use of traditional methods. That is why Data Mining techniques are being used to analyze these huge datasets, next to distributed systems that allow fast and smart processing of large amounts of data.

1.1 Motivation

The development of this thesis has been motivated by the researches of the Life Sciences (LS) department at Barcelona Super Computing Center. They have been studying deeply chromosome rearrangements in samples of cancer patients and noticed an interesting pattern: there are some occurrences where at least three different chromosome points rearrange together and share genetic information. An example of this kind of events can be seen in the schematic of fig. 1.4a. In this case, the genomic information of chromosome 1 at point a is exchanged with both point c of chromosome 11 and point b of chromosome 5. At the same time, this two points of this two chromosomes exchange the genomic information between them, generating the triangle pattern. In order to have this kind of patterns, LS researches have the hypothesis of the three locations of the chromosomes getting closer together (1.4b) and at some point, start breaking and interchanging their information.

The fact of this triangle pattern rearrangements not happening on a randomized environment means that they may be correlated with cancer
development. If a correlation is found, it would be a great improvement in personalized medicine [4].

1.2 Objectives

Considering the novelty of the research being done by the LS group, there are no tools and methodologies allowing us to find and work with this kind of complex rearrangements available. Previous to this thesis, Katsiaryna Krasnashchok presented some methods that could be used to develop a system capable of discovering complex patterns of chromosome rearrangements in an automatic manner. However, this methods were computationally hard enough allowing the analysis of only a few samples and did not have the enough precision for the triangles detection as we will see in further chapters. Therefore, the aim of this thesis is to provide a fully functional automatic system capable of processing complete datasets in a distributed environment.

Furthermore, once the system here developed is going to be applied to a genomic dataset from the International Carcinogen Genomics Consortium (ICGC), provided by LS department and make an exploratory analysis of the results.
1.3 Contributions

This thesis proposed an automatic method for detecting chromosomal rearrangements from genomic data with triangular forms. It uses unsupervised clustering based on KDE and Graph Mining techniques in a distributed environment for maximizing the time efficiency of the algorithm. A general overview of the architecture can be seen in fig. 1.5.

![Figure 1.5: Block diagram of the proposed architecture](image)

Furthermore, the proposed architecture has been applied to a genomic dataset and a potential bio-marker for sarcoma has been detected after an exploratory analysis on the results of the presented methodology.

1.4 Document organization

The rest of the document is organized as follows. Chapter 2 introduces the background knowledge about human genome needed to understand the work done in the thesis, next to the first approaches of the problem previous to this thesis. Chapter 3 reviews how big data techniques are being applied for solving different genomic problems. Chapter 4 presents the proposed architecture of the unsupervised clustering method presented in this work. Chapter 5 describes the graph mining algorithm and the exploratory analysis done after testing the proposed model. Chapter 6 summarizes the main conclusions of this master thesis and presents an outlook for future work.
Chapter 2

Background

This section describes the data used for the analysis and the approach of the Life Science research group. Furthermore, the starting point of the proposed system is also described at the end of the chapter.

2.1 Dataset

In order to compare the results in the most appropriate manner, the dataset used for this thesis is the same used at the Life Science department, obtained from the ICGC repository and pre-processed to display the anomalies on the provided sequences. The source dataset has been obtained from the PanCancer data repository [5] available online. The pre-processed data consists on tumor samples containing information such as the cancer type and the rearrangements occurring in that sample, next to other parameters that are not relevant for the current analysis, but will be in the future work.

Before describing how the data is structured, it is important to know some details about the Human Genome. As it has been introduced in chapter 1, Human Genome is composed by over 3 billion base-pairs. It is organized into 22 paired chromosomes, plus the X chromosome (one in males, two in females) and, in males only, one Y chromosome. It is important to mention that chromosomes are numbered by its size (fig. 2.1) and not by their location. Therefore, chromosome 1 is not next to chromosome 2: they are all mixed in the cell nucleus.

The data is structured in tables, where each table correspond with one sample. The structure of this tables can be seen in fig. 2.2a. Every row of the table describes one rearrangement (edges), where we have two chromosome breakpoints: one is the "source" and the other one is the "target". This connotations can be interchanged at any moment, since all the rearrangements are considered uni-directed, according to LS domain experts. These "source" and "target" points, are the locations that are interchanging information in a concrete rearrangement. These locations are described by two values: one is the chromosome number and the other is the base-pair position of that chromosome. The example rearrangements of fig. 2.2a can be seen represented in fig. 2.2b, where only the chromo-
some of interest are represented in order to simplify the complexity of the graph. As it can be seen in the graphic representation, it is an example of triangular chromosomal rearrangement.

![Figure 2.1: Chromosome size in base-pairs](image)

(a) Sample structure example. \{a1, a2, b1, b2, c1, c2\} correspond to base-pair positions inside their respective chromosomes.

(b) Sample example graphic representation

Since human genome contains over three billion base-pairs, it is highly unlikely to find two breakpoints happening in the exact same point. Because of it, the pattern analysis must be done using chromosome regions instead of exact locations. As it has been seen in fig. 2.2b, a1, a2, b1,b2 or c1,c3 are not the exact same point, but, they are close enough to each other to consider them the same region. Therefore, the challenge that must be solved before analyzing patterns is how big this regions must be (how close breakpoints must be in order to be joined).
2.2 Joining breakpoints

First, a clarification about which kind of triangles are we interested in, since that affects the joining protocol. Regarding LS domain experts, the relevant triangles are the ones that have at least two vertices in two different chromosomes. This means that we are not taking into account the intra-triangles. This intra-triangles are considered as chromotripsy \(^1\), since this event is highly probable to be due to multiple ruptures of a chromosome region that is not relevant for this study. Valid triangles will have 3 vertices in 3 different chromosomes or 2 vertices in one chromosome (far enough one of each other) and other in another chromosome.

2.2.1 Life Science approach

Life Science scientists decided to use a distance-based model using a fixed window to determine if two points belong to the same break or not. This methodology relays on the use of two parameters to detect the valid triangles: the Window or Inter-window and the Loop or Intra-window (2.3).

- **Window or Inter-window**: this parameter defines if two breakpoints of one chromosome belonging to two ends of an exchange between this chromosome and other chromosome(s) can be joined together. If the distance between the two breakpoints is less or equal than the inter-window size, those breakpoints can be considered belonging to the same region and can be joined.

- **Loop or Intra-window**: this parameter defines if two breakpoints of one chromosome belonging to two ends of the same exchange inside the chromosome itself can be considered as a part of a chained rearrangement or must be discarded or classified and chromotripsy. If the distance between those breakpoints is more or equal than the intra-window size, those breakpoints can be considered as a part of the chained rearrangement and therefore, the ends of exchange can be joined with the rest of the chain using inter-window.

Both window and loop size were fixed to create a conservative data set of triangles. There are no other studies about this kind of patterns so there is no measure to compare if the sizes are the right one or not, only the expert criteria. Therefore, LS experts decided to use a conservative approach

---

\(^1\)Chromotripsy or “Catastrophe phenomenon” is related to mitosis errors and involves a large number of rearrangements
in order to only join breakpoints that are really close and ensure the trustworthiness of obtained conclusions. The final values used were window = 2,000 base-pairs and loop = 14,000,000 base-pairs. This approach is a simple mechanism that allows to find triangles effectively but have some limitations.

- **Fixed window**: the use of an inter-window fixed value for determining if two breakpoints potentially belonging to a triangle must be join or not is a bit limited since it does not take into account the points surrounding them. In a scenario were two breakpoints of a potential triangle are separated by around 3,000 base-pairs but between them there are some other breakpoints that could indicate that all those points are involved in the same rearrangement will not be joined. Furthermore, these distances are likely to suffer from selection bias of it. Therefore, a solution were the distance can change depending on the volume of points of the region will be more interesting if based on the statistical properties of data and not in a fixed number.

- **Pre-defined shape**: this approach is exclusively designed for finding triangles. However, LS researches are not only interested in triangles but in all the potential patterns that could be found and proved to be relevant. Triangles are the main point of interest at this time of the research but a solution were other shapes will be also considered necessary for the later experiments.

- **Efficiency computation**: since this tool should be use to work with as much as data as possible, developing a system capable of using distributing computing techniques is highly important, in order to reduce the time of computation.
2.2.2 Starting point

Previous to this Master Thesis, Katsiaryna Krasnachok started to research different techniques to join the breakpoints and to perform the pattern search. Two of the methodologies proposed (KDE based clustering and Graph Mining) has also been used for the development of the architecture presented in this thesis. Therefore, they will be explained in deep in chapters chapter 4 and chapter 5. A short description is provided as follows.

- **KDE based clustering:** before searching for possible patterns, breakpoints must be joined. Based on the work done in [6], a Kernel Density Estimation (KDE) [7] of the breakpoints location is used to find peaks in density function and assign each breakpoint to its closest peak.

- **Graph Mining:** a modified version of VSIGRAM [8] is proposed. VSIGRAM is a subgraph mining algorithm that prunes the search tree by cutting the branches with low-frequency. It has been modified to prune without frequency limit, limiting the size of the patterns. In concrete, since the aim of the search is triangles, the number of edges established for the search is limited to 3.
Chapter 3

Related work

In this section we are going to review the current state of the art for some related research topics. First, we will review the rearrangements detection method that has been used for the data set used in this thesis. Then, we will review some of the most popular machine learning applications in genomics and afterwards, we will review which techniques are being used for processing large genomic data sets.

Rearrangements detection

The data provided by the PanCancer project is composed by sequenced tumour genomes, labelled by illness and clinical features. However, the dataset used in this thesis has also the information of the chromosome rearrangements. This information has been obtained using a Somatic Mutation Finder called SMuFin [9]. SMuFin is a free method developed by a collaboration between the Life Science and Computer Science (CS) department at BSC. It is designed to identify somatic variation on tumour genomes, the rearrangements, from the direct comparison with the corresponding normal genome of the same patient. Through a single execution is able to identifying somatic single nucleotide variants (SNVs) and structural variants (SVs) of any size.

Machine learning techniques

Genomics is a large field to be yet completely explored and many different Machine Learning techniques are being proposed and applied at this time. Some of the main applications are pathogenic prediction, gene finding, gene annotations, protein sequencing and modelling or recognizing patterns in DNA sequences. In the next paragraph, some of the state of the art methods for the different applications are presented.

In [10] they develop MP3, a tool that provides improved efficiency and accuracy to predict pathogenic proteins in both genomic and metagenomic datasets. It uses an implementation composed by the integration of Support Vector Machine (SVM) [11] and Hidden Markov Model (HMM) [12]. Another example is [13], where they present a gene predictor called SNAP, which models protein coding sequences in genomic DNA via a special-
ized Hidden Markov Model and Weight Array Matrices (WAM) for parameter estimations. For protein sequences analysis, the most popular method is the Basic Local Alignment Search Tool (BLAST) [14]: it performs local alignments using heuristics methods to produce quick results. Using heuristics methods can be of great interest for our future work, where we need to find different patterns. A recent method for recognizing patterns in DNA sequence is [15]. With it, short Dinucleotide Repeat Motifs (DRMs) are searched using Position Weight Matrices (PWMs) with probability of one for the appropriate nucleotide in each position. Another interesting work is T-KDE [16], a method for genome-wide identification of constitutive protein binding sites from multiple ChIP-seq data sets. They use a binary range tree algorithm to order the peak centers and convert them into subsets (terminal nodes) and afterwards, they apply KDE to estimate a density function from each terminal node. They use this estimation to divide the space into modal regions. Our clustering works very similar to this methodology, but without the need of using a binary range tree, applying the method to the breakpoints. We also use the local maximum (peaks) to divide the space and decide which breakpoints are joined.

**Approaches for data growth**

Lately and due to the recent advances in high-throughput technologies, researches have more data to work with, which opens the need of implementing Big Data methods and algorithms that can be applied to massive data sets. GeneMerge [17] is an example of applying Data Mining techniques to a data set. It returns a range of functional and genomic data for a given set of study genes and provides statistical rank scores for over-representation of particular functions or categories in the data set. In [18] they develop DN-Fold\(^1\), a Deep Learning approach for protein fold recognition. They use the network to predict if a given query-template protein pair belongs to the same structure fold, which is a problem that can be addressed as a binary classification problem simply labeling a protein pair which shares the same structural fold as ‘1’ and pairs from dissimilar folds are as ‘–1’. Deep learning is also used for gene prediction, as can be seen in [19]. They present a novel method ‘Meta-MFDL’ to predict metagenomic genes by fusing multifeatures and using deep stacking networks. All these Deep Learning techniques are the ones that we are going to test in the future work chapter 6.

Along with these techniques, efficient computation is also an impor-

\(^1\)http://iris.rnet.missouri.edu/dnfold
tant point for computing high volume data sets. In [20] they provide an overview of cloud computing and big data technologies, and discuss how such expertise can be used to deal with biology’s big data sets. In particular, big data technologies such as the Apache Hadoop project, which provides distributed and paralleled data processing is presented as a useful tool for it. Since we work with a large amount of sample breakpoints, we decided to use this technology to compute our data set in parallel and reduce the time consumption.
Chapter 4

Clustering Architecture

As it has been explained in section 2.1, joining the breakpoints is one of the principal challenges on this problem. The original rearrangement breakpoints of the dataset are highly disconnected due to the high number of base-pairs of the chromosomes. This makes the occurrence of patterns such as triangles very improbable. Furthermore, as the domain experts (LS scientist) have stated, two breakpoints that happens close enough are likely to be in the same interchange of information. Data is therefore unlabeled so we need to use an unsupervised method. The evaluation of the performance is going to be through domain expertise provided by researchers from LS.

The implemented method in this thesis to perform the joining of the breakpoints is based in using the KDE clustering of [6] modified by K. Krasnashchok. Some modifications has been done such as a distributed implementation to reduce the time consuming of this process and a two step processing including a cleaning step in order to increment the precision of the clustering. After the clustering, we will have the original rearrangements using the center of the cluster to which every breakpoint belongs to instead of the original exact position. This should reduce the granularity of data, allowing more connected observations, and hopefully it will discover patterns in the rearrangements that were hidden before. The block diagram of the presented architecture can be seen in fig. 4.4 and the different steps of it are going to be described in the following sections.

4.1 Data preparation

The first step before doing the clustering is preparing the data. This means transforming the dataset structure to a more readable one, also removing unnecessary data. The raw input data seen in table (fig. 2.2a) is transformed into a sample table where the rows are the breakpoints located in every chromosome, as shown in table 4.1.

This is done because the clustering is going to be done chromosome by chromosome, since it will make no sense to cluster breakpoints of different chromosomes (remember that chromosome numbers does not indicate connected chromosomes but an arbitrary nomenclature ordered by size).
Notice that the rearrangement information is discarded for the clustering, we are only dealing with the location of the breakpoints no matter the connection between them. This information will be recover at the end of the clustering.

### 4.2 KDE distributed clustering

The idea of this process is to join the breakpoints based on the density of the positions distribution and not in a fixed distance. Using a density estimation of the breakpoints as the starting point of the cluster allows us to still rely on the closeness of the breakpoints but also on the density. According to the domain experts, breakpoints in locations with a high density of breakpoints are likely to be in the same interchange of information.

The KDE distributed clustering process is done to every chromosome with breakpoints in every sample. The processing done to every chromosome is completely uncorrelated. This allows us to do it in a distributed environment using Apache Spark [21] programming model, where samples are processed in parallel. Apache Spark works following the structure of a distributed workers system, each work as an agent receiving chunks of data to be processed, and a central system orchestrating this multi-agent mechanism. The coordination step is done following the structure of master/slave with one central coordinator and many distributed workers.

Making the method computationally efficient is important since the data volume is high and it is supposed to increase with the pass of time. The method can be divided in three steps.

1. **Kernel Density Estimation**: based on the work done in [6], we are using Gaussian Kernel based on the normal distribution. At this point, we are just estimating the density of the breakpoints as it can be seen in fig. 4.1. During this process, one hyper-parameter must be set: the **bandwidth**. Increasing its value will lead us to bigger clusters and decreasing to smaller and more sparse clusters. In fig. 4.1
can be seen the differences between the usage of two bandwidth values.

2. **Peak locations**: once that the KDE is done, the next step is to localize all the peaks of the function. This peaks are going to be our cluster center and this is the location that is going to be use for all the breakpoints assigned to this cluster. In fig. 4.2 can be seen the KDE function with the peaks added in red.

3. **Breakpoints assignment**: finally, for every breakpoint its closest peak is found and is assigned to it. At the end of these three steps, we will have a table such as table 4.2 with the information of the center of the clusters and the breakpoints belonging to them for every chromosome and every sample.

![Kernel Density Estimation example](image)

(a) Bandwidth = 10.000

(b) Bandwidth = 2.000

Figure 4.1: Kernel Density Estimation example

After several experiments with different bandwidth values, we arrived to the conclusion that the precision of this method was limited for the size of the chromosomes. We observed that the density estimation was exactly
the same using any bandwidth equal or smaller to 1.000. The results provided with bandwidth = 500 show breakpoints being joined despite of being separated by distances greater than 7.000 base-pairs in almost 50% of the cases, being a 10% even greater than 100,000 base-pairs 4.3. Comparing with LS methodology (2.2.1), where they were using a window of size 2,000 and therefore all breakpoints of the same "cluster" were closer than 2,000 base-pairs, this numbers are clearly too big. Furthermore, doing the triangle pattern analysis that is explained in 5, we obtain twice the number of samples with triangles than LS team with their methodology. All this lead us to the conclusion that the clusters were too big and many

![Figure 4.2: Kernel Density Estimation example](image)

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>c1: [bp1, bp2]; c2:[bp3]</td>
</tr>
<tr>
<td>6</td>
<td>c1:[bp1]; c2:[bp2]</td>
</tr>
<tr>
<td>11</td>
<td>c1:[bp1, bp2, bp3]; c2:[bp4, bp5];...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 4.2: Post-KDE clustering sample table example
breakpoints being joined were not really interchanging information, so it makes no sense joining them. In order to obtain a more adequately join of the breakpoints, we decided to go cluster by cluster eliminating the points that were not relevant (no possible triangles) and then repeat the clustering process inside every cluster.

### 4.3 Solving lack of resolution: re-clustering

KDE clustering provides us the theoretic results that we are looking for joining the breakpoints but has shown a lack of precision. In order to solve it, we have decided to do a 2-time clustering: first we are doing the KDE clustering described in section 4.2 and then we are going inside every cluster and repeating a KDE clustering inside. However, before re-doing the clustering, it is better to make a cleaning of the non-interesting points. This way we avoid them to intercede in the result of the clustering. Therefore, we are doing a two steps mechanism here: **cluster cleaning** and **refining clusters by re-clustering**.

#### 4.3.1 Cluster cleaning

Having the clustering information in tables as table 4.2 and the rearrangements information as in fig. 2.2a, we can discern which breakpoints interchange information with the same cluster and which ones with a different cluster. Since we are looking for triangle events (or other kind of patterns in the future), points looping over the same cluster (two breakpoints edges are in the same cluster) will never form a triangle and there for we will leave them out of the analysis for the re-clustering. Thanks to this process we discard almost 42% of the original breakpoints (252,505 out of 614,624), what will make all the future computations faster and easier (less noisy).

#### 4.3.2 Re-clustering

Now that we only have the interesting breakpoints inside the clusters, we perform KDE clustering inside every cluster, in the same manner that was done before for every chromosome. From every chromosomes, we are

<table>
<thead>
<tr>
<th>Percentil</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>95</th>
<th>99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>100</td>
<td>558</td>
<td>1621</td>
<td>3531</td>
<td>7353</td>
<td>15k</td>
<td>31k</td>
<td>66k</td>
<td>147k</td>
<td>238k</td>
<td>446k</td>
</tr>
</tbody>
</table>

**Table 4.3: Percentils of intra-cluster breakpoints distance**
expecting to obtain one or more clusters from every cluster that we had before. In fig. 4.3 can be seen an example of the re-clustering process for three clusters of a chromosome. Since the size of the clusters is considerably smaller than the chromosomes and the amount of breakpoints has decreased, there is no need for making the algorithm distributed. We are working in a centralized environment.

Figure 4.3: Re-clustering process. In the top of the figure we can see how a KDE clustering is performed inside every cluster (after the cleaning process), with the aim of obtaining smaller clusters that fits better to the joining problem. In the bottom, it can be seen the final clusters of the chromosome after the re-clustering.

As it has been explained in section 4.2, KDE have only one hyperparameter that must be set, the **bandwidth**. This bandwidth has been set trying to fulfill the requirements of the domain experts. We aim to get a distance between breakpoints around 2.000 base-pairs. Furthermore, the number of samples with triangles for the same dataset obtained by their methodology (section 2.2.1) is 537. Therefore, we should expect to get a
Table 4.4: Re-clustering bandwidths comparison. For every bandwidth has been obtained the number of samples with triangles obtained, the number of samples with triangles obtained with both LS and the proposed method and the distance between breakpoints inside clusters at the 95 percentil. Final bandwidth decision has been also based on all the percentil values but only the most relevant for this case (95) is shown.

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th># Triangle samples</th>
<th># Common tr. samples</th>
<th>95 Percentil</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>391</td>
<td>380</td>
<td>545</td>
</tr>
<tr>
<td>100</td>
<td>430</td>
<td>416</td>
<td>701</td>
</tr>
<tr>
<td>250</td>
<td>510</td>
<td>492</td>
<td>1425</td>
</tr>
<tr>
<td>400</td>
<td>553</td>
<td>506</td>
<td>2123</td>
</tr>
<tr>
<td>600</td>
<td>564</td>
<td>506</td>
<td>3027</td>
</tr>
<tr>
<td>1000</td>
<td>593</td>
<td>512</td>
<td>4777</td>
</tr>
<tr>
<td>1500</td>
<td>614</td>
<td>512</td>
<td>6782</td>
</tr>
<tr>
<td>2000</td>
<td>639</td>
<td>513</td>
<td>8712</td>
</tr>
</tbody>
</table>

similar number. In table 4.4 can be seen a comparison of the results between different bandwidths in the re-clustering. The number of triangles has been obtained using the graph mining algorithm described in chapter 5.

The final selected bandwidth for the reclustering has been bandwidth = 400, since we obtain a number of sample triangles very similar to the ones of LS and the majority of them match. Furthermore, if we look to the distances between breakpoints of the same cluster, we can see how almost 95% of them are under 2,000 base-pairs, upper-limit that LS methodology uses. This fact is important since LS experts consider that only breakpoints that are really close must be closed in order to ensure the conclusions obtained have biological meaning.

### 4.4 Final Rearrangements

After the clustering process has been finished, the only step left to do before looking for triangles is to assign the new cluster points to the original breakpoints locations. Using the clustering information after the re-clustering in tables as table 4.2 and the original rearrangements information as in fig. 2.2a, we can easily assign to every original breakpoint, the value of the center of its cluster. The final clustering process can be seen in fig. 4.4.
1. **Fixed window**: basing our clustering architecture in a statistical method such as Kernel Density Estimation instead of using a fixed window allows us to join breakpoints taking into account not only the distance but also the quantity of ruptures occurring in the region of interest. Lone breakpoints should be clustered less than breakpoints happening in the same region, which are likely to be involved in the same rearrangements according to the domain experts.

2. **Pre-defined shape**: the focus of this thesis agrees with LS focus, which is finding rearrangement patterns with the form of triangles, since they think they could be directly correlated with the development of cancer cells. However, other rearrangement patterns such as squares or other kind of chains are likely to be of interest in the future. For this reason, clustering the breakpoints not taking into account if they are a potential triangle or not, allows us to apply any
kind of pattern search to the clustered data. Only breakpoints classified as chromotripsis has been discarded since they are not going to be involved in any pattern.

3. **Efficiency computation:** using a distributed algorithm for the main step of the clustering architecture allows us to cluster high-volume datasets efficiently.

To sum up, in this chapter we have presented an automatic clustering architecture based on unsupervised methods for joining breakpoints of genomic rearrangements datasets in a distributed environment. This method has been validated by LS domain experts. In the next chapter, the clustered data is going to be analyzed, searching for triangle patterns rearrangements and analyzing their occurrences.
Chapter 5
Exploratory Analysis

Now that the breakpoints have been joined, it is time to begin the pattern analysis. Using graph mining techniques, triangle patterns are going to be searched. Afterwards, a deep statistical analysis is going to be made, looking for anomalies that may indicate potential causes of cancer development.

5.1 Finding triangles

The definition of a triangle pattern has been given by the domain experts (LS research team) as three breakpoints interconnected, with at least 2 chromosome involved. If there are two breakpoints in the same chromosome, this breakpoints must be separated by 14,000,000 base-pairs. Other types of triangle patterns such as having the three break point vertices in a chromosome are leave out of the analysis because of its lack of biological interest for this study. Triangles are obtained following two steps. First, a distributed graph mining algorithm proposed by K. Krasnashchok in her thesis is going to be used to find all the triangle patterns. Then, using a couple of simple rules, the not interesting triangles are discarded.

5.1.1 Graph Mining

At this time of the methodology, every data sample can be interpreted as a graph where every cluster is a vertice and the translocations between breakpoints indicate the edges between clusters. Every vertice may have more than one edge since it is composed by several breakpoints. The task to do is finding sub-graphs with triangle form inside the graphs.

The algorithm proposed by K. Krasnashchok was based in VSIGRAM, presented by [8]. This algorithm uses canonical label as a sort of a hash code for each sub-graph. Labeling the graph’s vertices and edges with canonical labels, we assure the following: if two graphs have the same canonical label, they are isomorphic: structurally identical. Therefore, we do not have to search for duplicates using this technique. To adapt VSIGRAM to our particular problem, two main changes of the algorithm have been done.
• **Pattern size limited:** most sub-graph mining algorithm use a frequency threshold to prune the search, cutting the tree-branches with low frequency. However, in our case, low frequency cases are as interesting as high frequency. Therefore, we can not prune with this technique. What we can prune is the size of the patterns searched. **VSIGRAM** searches for all the possible sizes but we are only looking for triangles. Limiting the pattern size to 3, we are doing a size-based pruning that decrease considerably the search space.

• **Distributed:** as in the clustering process, the algorithm is implemented Spark programming model. Using parallel operations such as map, flatMap and reduceByKey.

This two adaptations makes the original algorithm more suitable for our problem. The dataset generated in the clustering step is now processed in the search of triangles following these steps: deduplicate edges, generate graph, mine sub-graphs, reduce similar graphs and valid triangles selection. All these steps can be seen in fig. 5.2.

**Deduplicate edges**

Since every cluster can group more than 1 breakpoint, it is likely to find clusters with more than one edge going to the same other cluster. This edges are therefore duplicated and must be deduplicated, eliminating all of them except one. An example can be seen in table 5.1.

<table>
<thead>
<tr>
<th>Cluster source</th>
<th>Cluster target</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>c2</td>
</tr>
<tr>
<td>c2</td>
<td>c3</td>
</tr>
<tr>
<td>c4</td>
<td>c3</td>
</tr>
<tr>
<td>c1</td>
<td>c2</td>
</tr>
<tr>
<td>c1</td>
<td>c2</td>
</tr>
<tr>
<td>c3</td>
<td>c1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cluster source</th>
<th>Cluster target</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>c2</td>
</tr>
<tr>
<td>c2</td>
<td>c3</td>
</tr>
<tr>
<td>c4</td>
<td>c3</td>
</tr>
<tr>
<td>c3</td>
<td>c1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 5.1: Sample rearrangements clustered before (left) and after (right) eliminating duplicates.

**Generate graph**

Once the duplicated edges have been deleted, next step is generating the full graph of the sample. This is done using the cluster centers as vertices
Table 5.2: The found sub-graphs are saved along with their canonical labels and edges involved. If a triangle is found twice, it should have both same canonical label and same edge’s hash. It will be removed from the table (bold rows).

Sub-graph mining

This step is done using parallelism: every vertex of the graph is processed at the same time. For a better understanding of the methodology, fig. 5.2 is a graphic representation of the different steps for one vertex. At every vertex, we look for all the possible connected paths of size 1. Then, this sub-graphs are the candidates for looking for all the possible connected paths of size 2. Finally, the process is repeated for the paths of size 3, the desired triangles. All the sub-graphs found are written in a table using canonical labels to identify the different patterns (table 5.2) . Edges belonging to a specific sub-graph are also saved using a unique hash in order to avoid counting the same pattern twice. All the sub-graphs obtained from all the vertices of the whole sample graph are saved in a common structure, where duplicated cases are easily eliminated due to the canonical labels and edges hashes.
Figure 5.1: Example graph and the different steps of the sub-graph mining algorithm for one of the vertices. This process is repeated for the four of them.

**Reduce similar patterns**

Once all the possible patterns of all the possible vertices have been discovered, the duplicates ones are deleted (keeping just one copy). Furthermore, similar patterns are group together (same canonical label), making easier to study different patterns afterwards.

**Valid Triangles**

The aim of this algorithm is finding triangles. Therefore, from the subgraphs discovered at last step, we are going to evaluate only the triangles (\{0:1,2; 1:0,2; 2:0,1\}). Since the form of the pattern is clear, we are going to work with the vertices instead of the edges. A triangle form by edges ‘c1:c3’, ‘c2:c3’ and ‘c1:c2’ will be saved as ‘c1:c2:c3’. Using this structure, is easier to identify if a triangle is valid or not.
Triangles can be divided in three classes: intra-triangles, inter-triangles and half-inter-triangles. Following the indications provided by LS research team explained in chapter 2, we can classify them and filter the non desired triangles.

- **Intra-triangles**: we called intra-triangles to those triangles having the three vertices in the same chromosome. These triangles are not relevant for the study, since they are likely to be a chromotripsy event.

- **Inter-triangles**: we called inter-triangles to those triangles having the three vertices in three different chromosomes. These are the main focus of study.

- **Half-inter-triangles**: we called half-inter-triangles to those triangles having two vertices in the same chromosome and another one in a different triangles. If the edges located in the same chromosome fulfill the condition of being separated by at least 14,000,000 base-pairs, we can consider a similar case to inter-triangles, and, therefore, a case of study. Otherwise they will be filter as intra-triangles.

![Figure 5.2: Blocks diagram of the graph mining algorithm.](image)

### 5.2 Exploratory Analysis

Valid triangles are already found and labeled. It is time to analyze how often they appear, how many and where do they appear.
5.2.1 Triangle analysis per Cancer type

First analysis that has been done is counting the number of triangles of every cancer type. Since the number of samples per cancer type is not the same and neither is the number of breakpoints of every sample, we can not compare the results in an absolute manner. Therefore, we have set two ratios: \( r_1 \), triangles per sample and \( r_2 \), triangles per breakpoints. With these two values we can obtain a more indicative information about if a cancer type has many or few triangles. In table 5.3 can be seen how values vary across cancer types. There are cases were we do not have any triangle at all (i.e. 30, LAML) and cases were most of the samples have triangles (i.e. 27, SARC, \( r_1 = 5 \)). There are also cases were we have a high number of triangles with a not so high number of total breakpoints (i.e. 10, LIHC). All this information will be useful to determinate if the results obtained in further analysis are valuable or not. Having 30 samples with triangles with a specific behaviour will be trustier than having only 2 samples with the same behaviour. At this time, we have observed how most of cancer types have multiple cases with rearrangements following the triangle pattern. This fact had been already detected by LS researchers and is now confirmed using our methodology, based on statistical analysis.

5.2.2 Triangle locations

Triangle patterns have been confirmed so the next step is studying its behaviour. First of all, the location of the breakpoints belonging to triangles are going to be study. Before focusing in the particularities of the different cancer types, we have study the global picture, doing a simple count of how many triangle breakpoints appear in every chromosome. In chapter 2 was explained that the number of the chromosomes are given due to its size, being 1 bigger than 5. This fact makes us expect more breakpoints located in the first chromosomes and less in the last. In fig. 5.3, blue bars represent the chromosome size and red bars the triangle breakpoints across all samples occurring in each chromosome. Red bars should show a decreasing behaviour such as blue bars (leaving out chromosome 23/X and 24/Y, the gender chromosomes that works differently since they do not appear always in the same quantity). However, we have peaks in chromosomes 12, 17 and 19 that are not supposed to be if the rearrangements happened randomly. This first anomaly seems to indicate that not only the apparition of triangle-based rearrangement may be involved in cancer development but also the location of these triangles may also be relevant.
<table>
<thead>
<tr>
<th>C.type</th>
<th>samples</th>
<th>s. w. tr.</th>
<th>bkpts</th>
<th>triangles</th>
<th>r1</th>
<th>r2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean value</td>
<td>62</td>
<td>14</td>
<td>6314</td>
<td>39.0</td>
<td>0.627</td>
<td>0.618</td>
</tr>
<tr>
<td>0</td>
<td>DLBC</td>
<td>7</td>
<td>1</td>
<td>380</td>
<td>1.0</td>
<td>0.143</td>
</tr>
<tr>
<td>1</td>
<td>PAEN</td>
<td>59</td>
<td>6</td>
<td>1643</td>
<td>8.0</td>
<td>0.136</td>
</tr>
<tr>
<td>2</td>
<td>RECA</td>
<td>66</td>
<td>4</td>
<td>1020</td>
<td>6.0</td>
<td>0.091</td>
</tr>
<tr>
<td>3</td>
<td>BOCA</td>
<td>54</td>
<td>26</td>
<td>12179</td>
<td>102.0</td>
<td>1.889</td>
</tr>
<tr>
<td>4</td>
<td>STAD</td>
<td>38</td>
<td>9</td>
<td>5769</td>
<td>35.0</td>
<td>0.921</td>
</tr>
<tr>
<td>5</td>
<td>UCEC</td>
<td>42</td>
<td>18</td>
<td>6099</td>
<td>42.0</td>
<td>1.000</td>
</tr>
<tr>
<td>6</td>
<td>THCA</td>
<td>27</td>
<td>1</td>
<td>165</td>
<td>1.0</td>
<td>0.037</td>
</tr>
<tr>
<td>7</td>
<td>LINC</td>
<td>28</td>
<td>6</td>
<td>1481</td>
<td>8.0</td>
<td>0.026</td>
</tr>
<tr>
<td>8</td>
<td>CESC</td>
<td>20</td>
<td>5</td>
<td>734</td>
<td>15.0</td>
<td>0.750</td>
</tr>
<tr>
<td>9</td>
<td>MALY</td>
<td>99</td>
<td>7</td>
<td>3428</td>
<td>20.0</td>
<td>0.202</td>
</tr>
<tr>
<td>10</td>
<td>LIHC</td>
<td>52</td>
<td>16</td>
<td>5599</td>
<td>79.0</td>
<td>1.519</td>
</tr>
<tr>
<td>11</td>
<td>READ</td>
<td>16</td>
<td>2</td>
<td>1076</td>
<td>2.0</td>
<td>0.125</td>
</tr>
<tr>
<td>12</td>
<td>LIRI</td>
<td>256</td>
<td>21</td>
<td>13358</td>
<td>62.0</td>
<td>0.242</td>
</tr>
<tr>
<td>13</td>
<td>HNSC</td>
<td>44</td>
<td>10</td>
<td>3019</td>
<td>18.0</td>
<td>0.409</td>
</tr>
<tr>
<td>14</td>
<td>SKCM</td>
<td>37</td>
<td>15</td>
<td>5305</td>
<td>82.0</td>
<td>2.216</td>
</tr>
<tr>
<td>15</td>
<td>GACA</td>
<td>33</td>
<td>14</td>
<td>4072</td>
<td>26.0</td>
<td>0.788</td>
</tr>
<tr>
<td>16</td>
<td>COAD</td>
<td>41</td>
<td>5</td>
<td>2182</td>
<td>12.0</td>
<td>0.293</td>
</tr>
<tr>
<td>17</td>
<td>BLCA</td>
<td>23</td>
<td>7</td>
<td>2909</td>
<td>21.0</td>
<td>0.913</td>
</tr>
<tr>
<td>18</td>
<td>PACA</td>
<td>237</td>
<td>48</td>
<td>20151</td>
<td>80.0</td>
<td>0.338</td>
</tr>
<tr>
<td>19</td>
<td>GBM</td>
<td>40</td>
<td>10</td>
<td>4544</td>
<td>50.0</td>
<td>1.250</td>
</tr>
<tr>
<td>20</td>
<td>LUSC</td>
<td>48</td>
<td>21</td>
<td>6582</td>
<td>45.0</td>
<td>0.938</td>
</tr>
<tr>
<td>21</td>
<td>CLLE</td>
<td>74</td>
<td>1</td>
<td>457</td>
<td>1.0</td>
<td>0.014</td>
</tr>
<tr>
<td>22</td>
<td>LICA</td>
<td>5</td>
<td>1</td>
<td>70</td>
<td>3.0</td>
<td>0.600</td>
</tr>
<tr>
<td>23</td>
<td>CMDI</td>
<td>15</td>
<td>1</td>
<td>131</td>
<td>1.0</td>
<td>0.067</td>
</tr>
<tr>
<td>24</td>
<td>KIRC</td>
<td>35</td>
<td>6</td>
<td>3065</td>
<td>12.0</td>
<td>0.343</td>
</tr>
<tr>
<td>25</td>
<td>BTCA</td>
<td>12</td>
<td>0</td>
<td>1126</td>
<td>0.0</td>
<td>0.000</td>
</tr>
<tr>
<td>26</td>
<td>ORCA</td>
<td>11</td>
<td>0</td>
<td>243</td>
<td>0.0</td>
<td>0.000</td>
</tr>
<tr>
<td>27</td>
<td>SARC</td>
<td>32</td>
<td>24</td>
<td>22129</td>
<td>163.0</td>
<td>5.094</td>
</tr>
<tr>
<td>28</td>
<td>KIRP</td>
<td>21</td>
<td>4</td>
<td>502</td>
<td>7.0</td>
<td>0.333</td>
</tr>
<tr>
<td>29</td>
<td>LGG</td>
<td>16</td>
<td>1</td>
<td>308</td>
<td>1.0</td>
<td>0.063</td>
</tr>
<tr>
<td>30</td>
<td>LAML</td>
<td>7</td>
<td>0</td>
<td>27</td>
<td>0.0</td>
<td>0.000</td>
</tr>
<tr>
<td>31</td>
<td>PRAD</td>
<td>214</td>
<td>60</td>
<td>23426</td>
<td>116.0</td>
<td>0.542</td>
</tr>
<tr>
<td>32</td>
<td>LUAD</td>
<td>36</td>
<td>4</td>
<td>4528</td>
<td>15.0</td>
<td>0.417</td>
</tr>
<tr>
<td>33</td>
<td>EOPC</td>
<td>61</td>
<td>8</td>
<td>2824</td>
<td>8.0</td>
<td>0.131</td>
</tr>
<tr>
<td>34</td>
<td>BRCA</td>
<td>204</td>
<td>75</td>
<td>40295</td>
<td>200.0</td>
<td>0.980</td>
</tr>
<tr>
<td>35</td>
<td>OV</td>
<td>112</td>
<td>50</td>
<td>21005</td>
<td>131.0</td>
<td>1.170</td>
</tr>
<tr>
<td>36</td>
<td>PBCA</td>
<td>187</td>
<td>1</td>
<td>2442</td>
<td>1.0</td>
<td>0.005</td>
</tr>
<tr>
<td>37</td>
<td>MELA</td>
<td>69</td>
<td>28</td>
<td>13441</td>
<td>105.0</td>
<td>1.522</td>
</tr>
<tr>
<td>38</td>
<td>KICH</td>
<td>26</td>
<td>3</td>
<td>978</td>
<td>13.0</td>
<td>0.500</td>
</tr>
<tr>
<td>39</td>
<td>ESAD</td>
<td>87</td>
<td>34</td>
<td>13865</td>
<td>69.0</td>
<td>0.793</td>
</tr>
</tbody>
</table>

Table 5.3: Statistic analysis of triangle frequency over cancer type. C. type corresponds with cancer type and s.w.tr. to samples with triangles.
Figure 5.3: Histogram of triangle breakpoints locations vs chromosome size. Both values have been normalized.

To discover the root of these peaks, we are going to go deep inside and study the locations of triangle breakpoints for every cancer type. This way we are going to find if these peaks occur in one cancer type, in several or in all of them. It is important to have always in mind table 5.3, because it tell us if the observed results of a cancer type is relevant or is due to the lack of samples.

**Cancer type analysis**

It can be seen in fig. 5.4 how every cancer type has his own unique histogram. Histograms of cancer types with low quantity of samples are easily visible since they only have breakpoints in a few chromosomes. However, cancer types with more than a few samples show a distribution across almost every chromosome. This is something that we were expecting. On the other hand, despite of having breakpoints across all chromosomes, cancer types seems to have tendency to accumulate breakpoints in different locations. For example, we can see how PACA (row 7, col 1) tends to have more breakpoints in last chromosomes. GBM and SARC in row 7 and 9, col 2 tend to have more breakpoints in chromosome 12. Almost none of the cancer types shows a flat decreasing behaviour that we were expecting if the breakpoints were happening randomly. With the aim of focusing our attention in cases less probable to be random (cancer types were breakpoints focus on one chromosome way over the random
occurrence), anomalies have been ranked.

**Ranking of anomalies**

The development of a measure of how out of normal are the locations of the breakpoints can be implemented comparing the size of the chromosome with the number of breakpoints in that chromosome, having both values normalized to the total length of chromosomes and the total number of breakpoints correspondingly. Dividing the breakpoints between the size, we obtain a number bigger than one if the breakpoints are occurring more than in a aleatory scenario and smaller if they are occurring less. As it has been said, there are cancer types with much more samples than others. Therefore, this number of samples is going to indicate if the data observed for a cancer type can be trusted or not if we do not have enough samples.

If a cancer type have less than 8 samples, is not going to be studied since the information is likely to not be enough to obtain any conclusion. In section 5.2.2 can be seen that GBM and SARC are the two cases with the bigger anomaly. This fact can be seen visually in 5.7. Furthermore, both cases have the anomaly in the chromosome 12, having more than 7 times the breakpoints they should have in that chromosome. This is clearly an out of normal case.

In terms of number of samples, it seems SARC results are better than GBM, having 24 samples against 10. From other cancer types, the anomalies that seems to be more legit are UCEC and PACA, having 18 and 24 samples both of them and a rank value over 4. Another relevant fact is that the chromosomes with higher breakpoints for every cancer type seems to be always the sames. Chromosomes 11,17,19 and 21 have the bigger number of breakpoints in 3 to 4 times while most of the other chromosomes never have an anomaly that big.
<table>
<thead>
<tr>
<th>Cancer</th>
<th>Rank</th>
<th>Chr</th>
<th>N. Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBM</td>
<td>7.88</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>SARC</td>
<td>7.00</td>
<td>12</td>
<td>24</td>
</tr>
<tr>
<td>UCEC</td>
<td>5.59</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>HNSC</td>
<td>5.36</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>EOPC</td>
<td>5.35</td>
<td>21</td>
<td>8</td>
</tr>
<tr>
<td>PACA</td>
<td>4.85</td>
<td>21</td>
<td>48</td>
</tr>
<tr>
<td>SKCM</td>
<td>4.23</td>
<td>11</td>
<td>15</td>
</tr>
<tr>
<td>GACA</td>
<td>3.90</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>LUSC</td>
<td>3.57</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>STAD</td>
<td>3.55</td>
<td>17</td>
<td>9</td>
</tr>
<tr>
<td>BOCA</td>
<td>3.52</td>
<td>19</td>
<td>26</td>
</tr>
<tr>
<td>BRCA</td>
<td>3.20</td>
<td>17</td>
<td>75</td>
</tr>
<tr>
<td>ESAD</td>
<td>3.08</td>
<td>17</td>
<td>34</td>
</tr>
<tr>
<td>LIRI</td>
<td>3.02</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>MELA</td>
<td>2.93</td>
<td>11</td>
<td>28</td>
</tr>
<tr>
<td>LIHC</td>
<td>2.66</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>OV</td>
<td>2.57</td>
<td>19</td>
<td>50</td>
</tr>
<tr>
<td>PRAD</td>
<td>2.16</td>
<td>21</td>
<td>60</td>
</tr>
</tbody>
</table>

Figure 5.5: Cancer types ranked by anomalies in their samples

Sample analysis

These cancer types having a higher amount of triangles for different chromosomes could be due to two different things: having one sample with many triangles in that chromosomes, or having many samples with a decent amount of triangles in that chromosomes. If we want to relate cancer development with the rupture locations of rearrangement following a triangle pattern, we need to find consistency across all samples of the same cancer type. In order to find out, the triangle breakpoints distribution is going to be observed for every sample of the cancer types.

In table 5.4 basic sample stats can be seen. In our search for consistency among samples of a cancer type, we have obtained the number of samples with a triangle breakpoint in the chromosome of interest. Furthermore, we have obtained the number of times that this chromosome has the highest amount of breakpoints.

The numbers indicate that consistency is only found among SARC samples, where 18 out of 24 samples have breakpoints in the chromosome 12 and 11 out of these 18 have the highest number of breakpoints. In the other
Table 5.4: Stats of cancer types with high anomalies. Anomalous chromosome, number of samples of cancer type, number of samples with triangles in anomalous chromosome and number of samples with maximum number of triangles in anomalous chromosome

<table>
<thead>
<tr>
<th>Cancer type</th>
<th>Anomalous Chr</th>
<th>N. samp.</th>
<th>N. w. tr</th>
<th>N. max tr</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBM</td>
<td>12</td>
<td>10</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>SARC</td>
<td>12</td>
<td>24</td>
<td>18</td>
<td>11</td>
</tr>
<tr>
<td>UCEC</td>
<td>19</td>
<td>18</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>PACA</td>
<td>19</td>
<td>48</td>
<td>11</td>
<td>8</td>
</tr>
</tbody>
</table>

For four cases, we have less than half of them having triangle breakpoints in the chromosome of interest. This fact indicates that the anomaly found is likely to be due to a couple of outlier samples.

Conclusions

As it has been seen in this exploratory analysis, we have identified that samples of cancer type sarcoma (SARC) have a very particular behaviour. Most of their ruptures occur in the chromosome 12. Some other cancer types have also particular behaviours but sarcoma is the most notable one. Other studies has already discovered translocations involving the chromosome 12 that directly correlate with the developing of sarcoma cancer variations such as liposarcomas [22]. Furthermore, it is known that this types of cancer suffer duplications of chromosome 12 [23]. This indicates that our unsupervised method has been able to discover correctly some known facts and can be used for discovering others.

Furthermore, between these breakpoints we can find a decent amount of rearrangements that form triangles between them. The importance of triangles has been explained in the chapter 2 and finding them in a chromosome where some other studies corroborate the importance of the breakpoints happening there, makes it an interesting discover for the domain experts. The initial hypothesis of this Thesis is based on LS hypothesis of triangles being directly related with the development of cancer. Finding them concentrated in a chromosome that has been proven to be involved in a concrete cancer type development improves the confidence over this theory.
Figure 5.4: Histogram of triangle breakpoints locations vs chromosome size per cancer type. Both values have been normalized.
Figure 5.6: Histogram of triangle breakpoints locations vs chromosome size per cancer type. Sorted by degree of anomaly. Both values have been normalized.
Figure 5.7: Histogram of triangle breakpoints locations vs chromosome size for SARC samples.
Chapter 6
Discussion

Contributions

The first contribution of the thesis has been the unsupervised clustering method for joining the breakpoints of a sample. These clustering is based in the breakpoint distribution across samples and has been proved to perform the joining correctly, showing results validated by domain experts. Since this is an automatic method, it allows us to process any number of samples without any need of human interaction. Furthermore, it is prepared to work in a distributed environment using Apache Spark technologies, computing the samples in parallel to reduce the computing time.

Next to the clustering, a graph mining algorithm has also been proposed, with the aim of finding patterns inside the clustered data. This algorithm is designed to find triangles since is the aim of this work but could be easily adjusted to find any other pattern. The only drawback would be that is an iterative method and as hard is the pattern (as more edges it has), as longer the computational time will be. In order to make it more efficiently, it has been designed also to be computed in Apache Spark. This time, instead of computing the samples in parallel, all the vertices of the sample graph are computed at the same time.

Both the clustering and the graph mining algorithm are based on the methods proposed at the master thesis of K. Krasnashchok. However, the original clustering algorithm was a simple non-distributed KDE clustering method, having neither the re-clustering step nor the distributed architecture. Original graph mining algorithm was also modified to obtain the specific triangles that fulfill the conditions given by LS researches. Moreover, the original methodology (both clustering and graph mining) was not prepared for computing full data sets in an automatic manner and therefore, it could only be tested with a few samples. The proposed architecture in this thesis is fully integrated and ready to process any amount of data.

In fact, the whole architecture has been applied to a genomic dataset and patterns detected have been submitted to an exploratory analysis. Through this analysis, triangle rearrangement have been found to happen consistently in chromosome 12 of sarcoma cells. This results matches the discovered done in other studies where different types of liposarcomas were
annotated to have special behaviours in this chromosome 12 [22][23][24], corroborating the usefulness of the developed architecture and proposing a potential bio-marker for this type of cancer.

**Future work**

In order to analyze not only the location of the triangle breakpoints but also how this location are combined within triangles, the first implementation that is being done is using Induction of Associative Rules [25]. This will allow us to study the statistic relationships between the vertices of the triangles.

On the other hand, the work of this thesis is based on the hypothesis of a direct relation between cancer development and rearrangement with triangle shape. Life Science researches has chosen the triangle shape as first shape of study since is the simplest pattern that can be easily identify. However, there is no certainty about it. That is why it was important to develop a method able to detect not only triangles but also more complex patterns.

Finding complex patterns in a graph doing a heuristic search is a feasible task but can be very time consuming if the size of the pattern begins to grow. Furthermore, it is a harder task when the pattern is unknown. Our focus know is to develop a system able to find more complex patterns in a more efficient way. Using neural network models to drive the search can reduce the pattern search. Being able to indicate whether a pattern is relevant and consistent will permit the search to discard noise events. Using discovery and predictions models, we will be capable to discriminate relevant patterns from noise with higher confidence on results.

More concretely, we have though on the he application of Convolutional Neural Networks (CNN) on the detection of relations between events, by learning the probability of events and patterns in graphs and samples. Since this kind of neural networks are usually used for detecting elements on images, we can treat the samples as a graph of rearrangements and take advantage from detecting elements for discrimination of noise events, as NNs focus on learning from behaviour models instead from random events. Another kind of network such as Recurrent Neural Networks (RNN) can also be used, this one as a generative model, to learn the probabilities of patterns in graphs given prior graph structure and patterns. We could use them to forecast evolution on patterns and extracting probabilities of sub-patterns given prior patterns, learning which pattern components lead to new ones.
References


