CRUK

Jorge Lemos Portela

July 16, 2021

Contents

1 Introduction

1.1 Mutational processes and the concept of mutational signatures

Throughout our lives our genome suffers many mutations. As time passes, there are many factors that lead to mutations in the genome like smoking tobacco or excessive exposure to ultraviolet light. Some of these mutations will not have a significant effect in our health but some others may lead to serious diseases, including cancer. Thus, understanding the biological processes generating these mutations is crucial.

With the technologies that we have nowadays we can compare the genome from the normal tissue of a patient with the sequenced genome of a cell of interest. This allows us to see all the aberrations present in that cell. However, timing these mutations can be challenging, and so is understanding the cause behind them.

Given the data of the mutations from a group of cancer patients, the signature analysis tries to find patterns in the mutations with the aim of finding processes that contribute to the overall set of mutations. Signature analysis consists mainly in three steps. The first step is feature extraction. In this step, we need to define the features of the genome that will be studied and extract the data about these features for each sample. The second step is signature identification. With the help of mathematical methods, the data about the mutations collected in the previous step is clustered into signatures. Moreover, each sample will be defined as a mixture of these signatures. The last step is signature assignment, where the signatures will be analysed and we will look for possible relations between these signatures and known biological processes or use them to stratify patients with the goal of finding more personalized treatments.

In general, the first step of signature analysis is crucial, since the way we define the features will determine which signatures we are able to extract. Once we have made a final decision for the features, we also need to decide on a method for the signature extracture and identification step which, as we will see later, may make a difference in the final results. Finally, the last step is the most challenging, as we need to find possible associations with biological processes or clinical data and there is no clear guidance for this. Nevertheless, this step is the one of biological interest as it may allow us to better understand the causes behind each mutational signature.

1.2 Signature analysis for SNVs and CNAs

In 2013, one of the first papers in this area was published [1]. In this article, they analysed the Single Nucleotide Variants (SNVs) in a collection of samples from different types of cancer. By following the steps described above, they created a catalogue of 30 distinguishable signatures that were later associated with biological processes.

The collected information about the SNVs was the substitution at the variant position (6 possibilities) and the nucleotides at the previous and posterior positions (16 possibilities). The data was not strand-specific so they had 96 possible categories for each SNV. The first step of feature extraction consisted in counting, for each sample, the number of mutations belonging to each of the categories. With this information, a sample by component matrix was created. For the next step of signature extraction, the Non-negative Matrix Factorization (NMF) algorithm was used to divide the sample by component matrix in a signature by component matrix and a sample by signature matrix. This allow us to understand the properties behind each signature and to see the signature exposures in the samples. Once the mutational signatures were defined and each sample defined as a combination of signatures, Generalised Linear models (GLMs) and Kolmogorov–Smirnov tests were used to relate the signature to causes. This causes include ageing, use of tobacco, specific mutations of interest...

The promising results from this paper increased the general interest in the study of mutational signatures. As we have seen, [1] gave a possible method to define and extract the signatures. But there were many other possibilities to explore. Moreover this paper studied 30 different cancer types extracted general signatures for all of them.

Later on, in 2018, [2] introduced the concept of copy number signatures, we will refer to the procedures in this paper as *Geoff's methods*. There are some types of cancer that are copy number driven, so in these scenarios it is more useful to have copy number signatures. The idea of this article was to adapt the steps for SNVs that were described above to the case of copy number aberrations in the genome.

The basis of the procedure for the analysis of copy number signatures will be similar to the one for SNVs introduced in [1], but there will be some additional complications. For starters, in the case of SNVs the features were clear, the interesting position was the variant baseposition, and then there was information about inmediate 3' and 5' bases to that position. However, Copy Number Aberrations (CNAs) can affect pieces of the genome of different sizes, there are multiple possibilities for the number of copies and also many other characteristics about the neighbor regions to explore. This implies that, for CN signatures, there will be some challenging decisions to be made even for the step of feature selection.

Let us explain the main steps of Geoff's method for CN signature analysis, which can be seen in Figure 1. First, the absolute copy number is derived from the shallow whole genome sequencing data. Next, the features to study need to be chosen, as was mentioned above this is a crucial step. The selected features define the results of the procedure. We will be looking for biologically meaningful or clearly distinctive features in the copy number states. The ones selected by Geoff can be seen in Figure 1. The first feature is the Breakpoint count per 10Mb. This counts the number of changes of CN value in each 10Mb segment along the genome. Next we have the Copy number which, as its name indicates, saves the information about the number of copies of each segment in the genome. The next feature is the Copy number change point, which measures the size of the change between CN changes. Then, we have the Breakpoint per chromosome arm, which again looks at the breakpoints count but this time in each chromosome arm. Moving on, we have the length of oscillating CN, that measures, for a given segment with $CN = c$ are the number of intermediate seg-

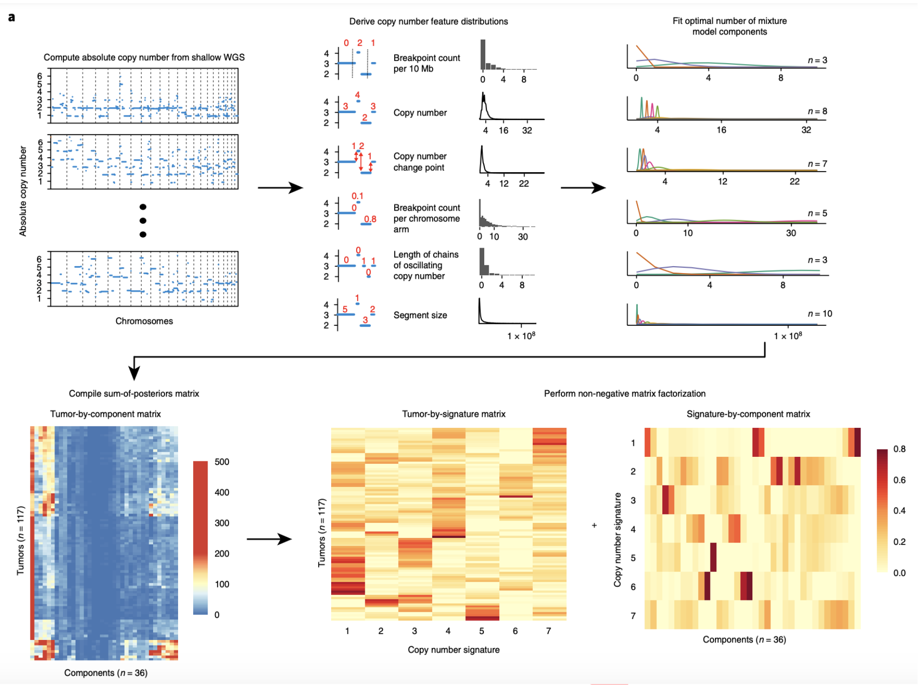

a
Compute absolute copy number from shallow WGS
Absolute copy number
Chromosomes
Derive copy number feature distributions
Breakpoint count per 10 Mb
Copy number
Copy number change point
Breakpoint count per chromosome arm
Length of chains of oscillating copy number
Segment size
Fit optimal number of mixture model components
n = 3
n = 8
n = 7
n = 5
n = 3
n = 10
1 × 10⁸
Compile sum-of-posteriors matrix
Tumor-by-component matrix
Tumors (n = 117)
Components (n = 36)
Perform non-negative matrix factorization
Tumor-by-signature matrix
Copy number signature
Signature-by-component matrix
Copy number signature
Components (n = 36)
500
400
300
200
100
0
0.8
0.6
0.4
0.2
0.0

Figure 1: *Picture extracted from [2] that explains the procedure to generate copy number signatures from the dWGS data.*

ments before there is another segment (if there is any more in that chromosome) with $CN = c$. Lastly we calculate the segment size, that measures the length of each CN segment.

Once the features are selected, the genome-wide distributions of these features are computed. The assumption is that each of these distributions can be seen as a mixture of either Poisson or Normal distributions. Thus, using a method of mixture models, the distributions for the features are divided into combinations of Poisson or Normal distributions, which will be refered to as the components of the feature.

We are interested in having a sample by component matrix that would reflect the exposure of the components in each sample. For each CNA event, there is a posterior probability that this event belongs to each of the components. This posterior probabilities are summed for all the CNA of each sample, creating the sample by component matrix, where the rows are the samples and the columns are the intensities of each component in that sample (see Figure 1).

Once we have the sample by component matrix, we need to decide the number of signatures that we are going to be clustering the samples in. This is a delicate decision. Besides the mutations that we are interested in, there is always some noise in the experiments, either from mutations that are unrelated to the cancer or from experimental noise. A higher number of signatures will focus on small details of the samples and this will not be ideal when the noise in the samples is relatively high. Also, we need to keep the final goal in mind, which is to relate the signature to possible causes. This will be more complicated if we have a high number of signatures or if they are highly correlated. Thus, there will not be a clear ideal number of signatures and we will just need to find a sufficiently good number of signatures to collect the most important information from the samples, that does not give much importance to the noise and which can be correlated later with possible causes.

Taking all of this into account the Non-negative Matrix Factorization (NMF) was used to decompose the sample by component matrix into a sample by signature matrix and a signature by component matrix. The NMF package in R [5] allows us to do this decomposition and it also calculates the cophenetic, dispersion, silhouette and sparseness coefficients for the resulting matrices. A range of possible signatures from 3 to 12 was selected and the NMF was used for each of this possibilities. The resulting coefficients were used to decide that the optimal number of signatures that yielded both high stability and spareseness was 7 signatures. We can see in Figure 1 the resulting matrices for this number of signatures.

So far, there were three main steps that were mentioned during the explanation above. The features' selection, the use of mixture models to create the sum of posteriors matrix and the use of NMF to divide the sample by component matrix into the sample by signature and signature by component matrices. After the NMF is performed, the goal is to relate the signatures to clinical data or biologically meaningful processes. The information of the signatures depends on the features that we choose in this first step, so we will also need to keep this in mind when we decide the features. The details about the NMF and the mixture models can be found in the appendix.

1.3 Shiraishi's alternative to NMF

After [1] was published in 2013, there were many other researchers that tried to develop new alternative methods for signature analysis. For the step of signature identification both [1] and [2] used NMF in order to obtain the signature by component and the sample by signature matrices. Since there is no guarantee that NMF is the optimal way to extract the signatures, [3] presented a new method for signature identification in the case of SNVs. This method is based on Expectation Maximization (EM) and has some advantages with respect to NMF, such as improved stability of statistical estimation, and easier implementation of additional components for the features. We will refer to the algorithms from [3] as *Shiraishi's methods.*

The goal of this article was to present new methods for modelling, identifying and visualizing mutation signatures. These methods were build with the idea of improving the previous procedures for the study of SNV signatures. They allowed both to simplify the previous models by reducing the number of parameters and, at the same time, also allowing to explore a higher number of flanking bases with resepect to the SNVs. This improves both sensitivity and robustness of inferred signatures. The overall framework of their work was based on probabilistic models that are closely connected with mixed-membership models, and it is known as Latent Dirichlet Allocation (LDA).

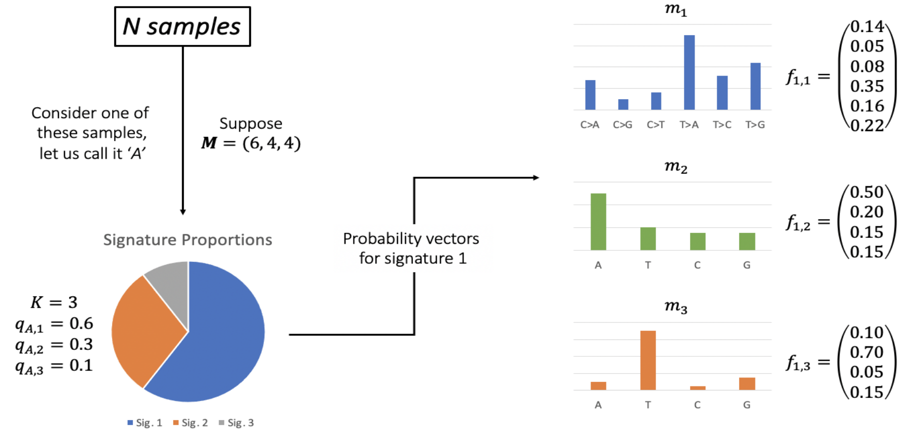
The main motivation for Shiraishi's paper was that the existing methods have two important limitations, caused by the fact that they use an unconstrained model for each mutational signature. First, although it may seem counterintuitive, using an unconstrained reduced flexibility. This is because the price of using such models is that one must limit the domain of mutation signatures considered. Moreover, most previous studies only collected information about one flanking base 3' and 5' if the SNV, which may be insufficient for the purpose of association with possible causes. These models took this approach because, in the unconstrained model, incorporating the more distal bases into the signature substantially increases the number of parameters, making estimated mutation signatures unstable. Secondly, and just as important, the unconstrained model means that each signature is a probability distribution in a high-dimensional parameter space, which can make signatures difficult to interpret.

One of the most important assumptions in [3] is the independence assumption accross features. This assumption is the reason why we can easily increase the number of flanking bases to study next to the SNV. For example, if we choose to study two flanking bases before and after the SNV and we do not assume independence, then there are $4 \times 4 \times 6 \times 4 \times 4 = 1536$ possible components (4 possibilities for the bases and 6 for the type of substitution). On the other hand, if we assume independence accross the features, then we have $4+4+6+4+4 = 22$ components. Although assuming independence among features in a signature may initially seem unnatural, in [3], they argue that its use is analogous to position weight matrix models, which have been highly successful for modelling transcription factor binding motifs.

Same way as in [1], we are looking at SNV signatures and we collect information about the type of substitution that took place and about the inmediate 3' and 5' bases next to that position. As was said above, Shiraishi's method allows to look for more than just one base before and after the SNV. This is an advantage which allow us to explore a higher number of components. However we need to keep in mind that we will have to relate the signatures to possible causes later. So we need to fit an optimal number of components that will encompass as much information as possible while at the same time being easily interpretable. A final choice of two bases before and two after the SNV was chosen for the paper. Also, they introduced an additional parameter to indicate in which strand the SNV happenned.

Let us explain Shiraishi's method now. Suppose each mutation has L features, $\mathbf{m} = (m_1, m_2, ...m_L)$, where each feature m_l can take M_l discrete values. Also let $\mathbf{M} := (M_1, M_2, ...M_L)$. In the case described above of an SNV and the 2 flanking positions before and after the variant this yields $\mathbf{M} = (6, 4, 4, 4, 4)$. Now suppose we have N sampled cancer genomes, and let J_i denote the number of mutations in the i-th cancer genome. For $i \in \{1, ...N\}$ and $j \in \{1, ...J_i\}$, we define $\mathbf{x}_{i,j} = (x_{i,j,1}, ...x_{i,j,L})$ as the observed mutation feature vector for the j-th mutation of the $i - th$ cancer sample, where $x_{i,j,l} \in \{1, ..., M_l\}$.

Moreover, assuming we are considering K signatures, let $\mathbf{q}_i = (q_{i,1}, ...q_{i,K})$ denote the proportion of signature k in sample i. Further, each mutational signature is characterized by parameter vectors $F_k := (f_{k,1}, \ldots, f_{k,L})$, where $f_{k,l}$ is a probability vector for the l-th feature in the k-th signature. An representation of these variables can be found in Figure 2. The figure shows an example where we consider only the next and previous positions to the SNV (so $\mathbf{M} = (6, 4, 4)$)



N samples
Consider one of these samples, let us call it 'A'
Suppose $M = (6, 4, 4)$
Signature Proportions
$K = 3$
$q_{A,1} = 0.6$
$q_{A,2} = 0.3$
$q_{A,3} = 0.1$
■ Sig. 1 ■ Sig. 2 ■ Sig. 3
Probability vectors for signature 1
m_1
C>A C>G C>T T>A T>C T>G
$f_{1,1} = \begin{pmatrix}0.14\\0.05\\0.08\\0.35\\0.16\\0.22\end{pmatrix}$
m_2
A T C G
$f_{1,2} = \begin{pmatrix}0.50\\0.20\\0.15\\0.15\end{pmatrix}$
m_3
A T C G
$f_{1,3} = \begin{pmatrix}0.10\\0.70\\0.05\\0.15\end{pmatrix}$

Figure 2: *Representation of an example to show the variables described by Shiraishi. The plot gives a visual representation of some of the variables from the model, but it is not a guidance of the steps to follow (see next Figure)*

and can be used to better understand the aboved described variables.

Shiraishi's model for the observed mutations $x_{i,j}$ in each cancer sample can now be described as a two-step process.

1. Generate $z_{i,j} \sim \text{Multinomial}(\mathbf{q}_m)$ where $z_{i,j} \in 1, ..., K$ denotes the unobserved mutational signature that caused the j-th mutation in the i-th sample.

2. For each $l \in \{1, ...L\}$ generate the observed mutation values as $x_{i,j,l} \sim \text{Multinomial}\left(\mathbf{f}_{z_{i,j},l}\right)$. Thus,

$$P(x_{i,j,l} = m | z_{i,j} = k) = f_{k,l,m}.$$

The key parameters in this model are the membership proportions for each sample, $\mathbf{q}_i$, and the mutation signature parameters, $\mathbf{F}_k$. These parameters are unknown and need to be estimated by maximizing likelihood using an Expectation Maximization (EM) algorithm with the available mutational data.

Let $g_{i,\mathbf{m}}$ denote the number of mutations in the i-th sample that have mutation feature vector $\mathbf{m}$. In the Expectation (E) step of the EM algorithm we create a function for the expectation of the log-likelihood, $\Theta_{i,j,\mathbf{m}}$, evaluated at the current estimates:

$$\Theta_{i,k,\mathbf{m}} = \frac{q_{i,k} \prod_{l=1}^{L} f_{k,l,m_l}}{\sum_{s=1}^{K} q_{i,s} \prod_{l=1}^{L} f_{s,l,m_l}}$$

And in the Maximization (M) step, we update the parameters $\{f_{k,l}\}$ and $\{q_{i,k}\}$, with the goal of maximizing the log-likelihood:

$$f_{k,l,p} = \frac{\sum_{\mathbf{m}:m_l=p} g_{i,\mathbf{m}} \Theta_{i,k,\mathbf{m}}}{\sum_q \sum_{\mathbf{m}:m_l=p} g_{i,\mathbf{m}}, \Theta_{i,k,\mathbf{m}}},$$

$$q_{i,k} = \frac{\sum_{\mathbf{m}} g_{i,\mathbf{m}} \Theta_{i,k,\mathbf{m}}}{\sum_s \sum_{\mathbf{m}} g_{i,\mathbf{m}} \Theta_{i,s,\mathbf{m}}}.$$

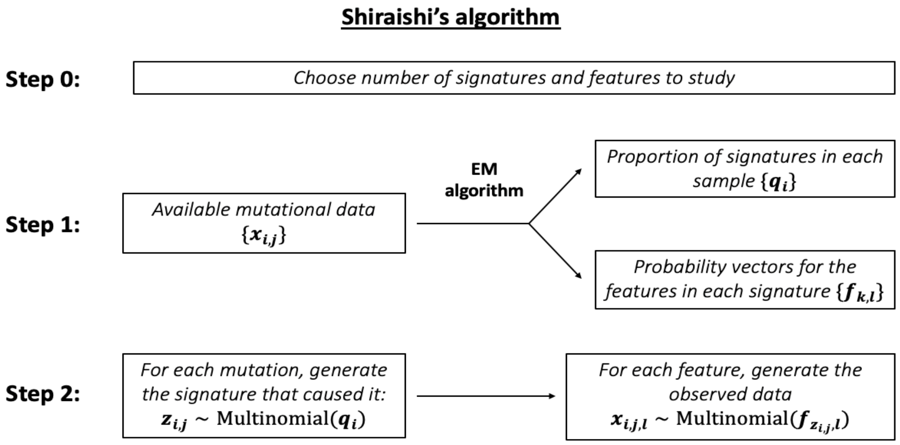

Shiraishi's algorithm
Step 0:
Choose number of signatures and features to study
Step 1:
Available mutational data $\{x_{i,j}\}$
EM algorithm
Proportion of signatures in each sample $\{q_i\}$
Probability vectors for the features in each signature $\{f_{k,l}\}$
Step 2:
For each mutation, generate the signature that caused it: $z_{i,j} \sim \text{Multinomial}(q_i)$
For each feature, generate the observed data $x_{i,j,l} \sim \text{Multinomial}(f_{z_{i,j},l})$

Figure 3: *Summary of Shiraishi's algorithm divided in three different steps.*

In Figure 3 we can see a representation of the algorithm. First, the number of signatures and the mutational features to study are chosen. Then, the Expectation Maximization (EM) algorithm is used to find the proportion of the signatures present in each sample and the probability vectors for the features. The algorithm finds these parameters by likelihood maximization. Lastly Multinomial distributions are used to generate, for each mutation, the signature that caused them, $z_{i,j}$, and for each feature the final observed data $x_{i,j,l}$.

1.4 Motivation of this work

There have been numerous papers about signature analysis in recent years. The different data sets and choices during the process give rise to different results. For the purpose this work we are particularly interested in two of these papers, [2] and [3].

On the one hand, [2] focused in ovarian carcinoma. This is a copy number driven type of cancer so the idea was to adapt the steps for SNVs that were described in [1] to the case of copy number aberrations in the genome. For the step of signature identification they used NMFwhich had been previously used for the case of SNVs by [1]. On the other hand, [3] presented a new method for signature identification in the case of SNVs as an alternative to NMF. This method was based on Expectation Maximization (EM) and had some advantages with respect to NMF such as improved stability of statistical estimation, and easier implementation of additional components for the features.

The goal of this paper will be to combine the ideas of [2] and [3]. We will analyse the same data that was used in [2] for the study of high-grade serous ovarian (HGSOC). The difference is that we will adapt the methods of [3] from the case of SNVs to the case of CNA in order to re-derive the signatures and then compare them with the current signatures that were identified in [2].

In the first section we will explain how the methods from [3] were adapted from the case of SNVs to the case of copy number analysis. Then, the second section will present the main results once we use this method with the data of ovarian cancer samples. This results will follow the first steps described above, from the feature extraction, to signature identification. Moreover, the results will be compared to the previously defined signatures and linked to other clinical information. The next section will explore other possibilities when we make different choices and combines some of the alternatives to create a more complex yet more complete scenario. Finally there will be discussion of the results and some comments on possible future work.

2 Methods

The aim of our work is to adapt Shiraishi's algorithm to copy number signatures and then analyse the results. In this section we will explain all the different steps that we had to follow in order to do this adaptation and all the choices we had to make along the way to get the final results. Some of this choices are crucial for the final results and a different selection would have lead to a potentially different outcome. Nevertheless, there is a reason behind each choice that was taken, and all of them are justified along this section.

We will start by explaining where the data comes from and how it was edited to get to the final version. Then, we will introduce the concept of a '*segment*' in our work, which will be the equivalent for an SNV and its neighbor regions in the case of Shiraishi. These segments will have a particular size which will also need to be defined. Moving on, we will select the features to study in these segments and, for the purpose of the algorithm, these features will need to be categorized into components. Once we have the components defined, we will need to select the optimal number of signatures that we will study to get our first results.

2.1 The data

In this work, we will use a subst of the data from [2], where they generated absolute copy number profiles from 253 primary and relapsed HGSOC samples from 132 patients in the British Translational Research Ovarian Cancer Collaborative (BriTROC-1) cohort using low-cost whole-genome sequencing (sWGS; 0.1×) and targeted amplicon sequencing of TP53. They further used other cohorts for validation, but for the purpose of our work we will just consider the BriTROC-1 cohort, which was the basis of their study.

Absolute copy number calling was performed from the sWGS data and the samples were classified depending on their quality. They used three different levels to describe the quality of the profiles. Out of the initial 132 patients, there were 91 patients whose samples were classified as the highest level of quality, and these are the ones we will use for our analysis.

The first step of our work was to approximate the copy number values into natural numbers. This implied the choice of a threshold to decide the final number of copies in each case. We decided to round the decimal copy number values to its closest integer. The reason behind this was to have a more realistic scenario without decimal number of copies.

2.2 The definition of a segment

Now that we know how Shiraishi's algorithm works for the case of SNVs, our goal will be to adapt it to the case of copy number signatures. As we have previously mentioned, there will be many new facts to consider. In [2], they showed a first method for the study of copy number signatures that was introduced in the first section of this work. Some parts of this method can be used or adapted to our work, but most of the steps will need to be rebuilt. The first difference between Geoff's and Shiraishi's methods is that Shiraishi's collects information about individual aberration events, while Geoff's studies the aberrations in general for each sample.

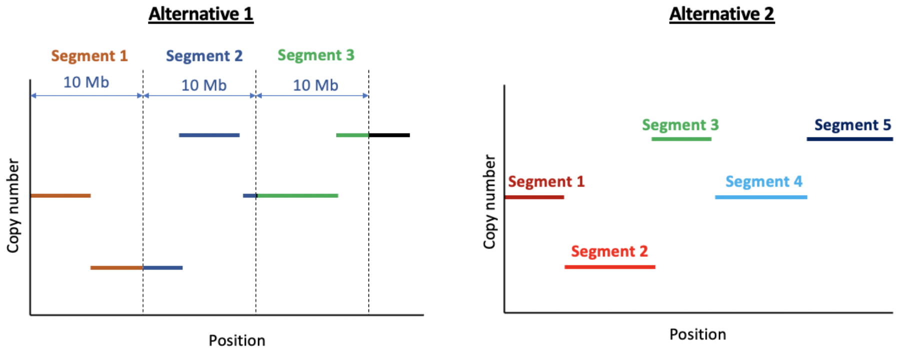


Figure 4: *The two strategies to define copy number aberration segments.*

In the case of SNVs it was simple to use Shiraishi's method, as the individual aberrations were the positions where the variant occurred. However, for copy number there are different ways to define an individual copy number aberration event. Here we will study two possible ways of defining copy number aberration segments. These two possibilities can be seen in Figure 4. The picture on the left shows the first alternative, where we divide the genome in segments of 10Mb and study the features in each of these pieces. The other alternative (shown on the right) consider CN segments as pieces of DNA with the same copy number. Alternative 1 has the advantage that we can study features such as breakpoint count or mean CN per 10Mb. On the other hand, the main advantage of Alternative 2 is the study of segment sizes, where segments are defined as connected pieces in the genome with the same number of copies.

This is the first important decision that we will need to make. Since both methods have different perks, we will study both alternatives before making a final choice. However, for the moment we will only consider the scenario of the Alternative 1, where the genome is divided into 10Mb segments.

2.3 Choosing the size of the segments

Once we have made the decision of dividing the genome in 10Mb segments we should study the possibility of other sizes for the segments. The optimal segment size will need to be large enough to capture the necessary information about the features while, at the same time, short enough to have a high amount of segments for the algorithm to perform well.

Another important fact is that there will be part of the DNA whose information is not considered. For each chromosome of size C and given a selected size S for the segments. The chromosome will be divided into $Q = \text{floor}(C/S)$ segments. But the remaining $C - Q \times S$ Mb will be lost. For example, if we have a chromosome of length $C = 85$Mb, and we are working with segments of size $S = 10$Mb, then the chromosome would be divided into $Q = 8$ segments and 5 Mb will be lost. The amount of information that is lost is bounded by the size of the segments, so this is another fact to consider when choosing the optimal size.

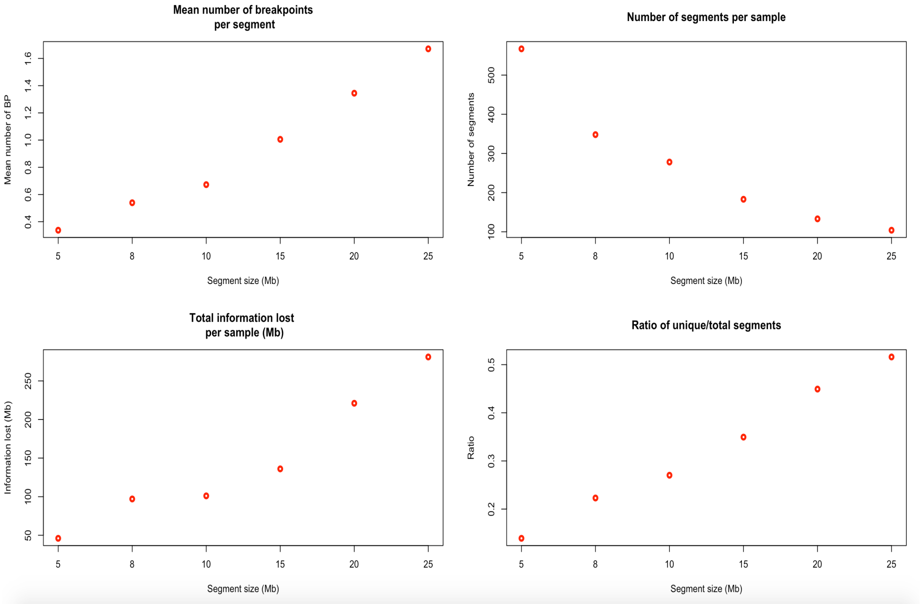


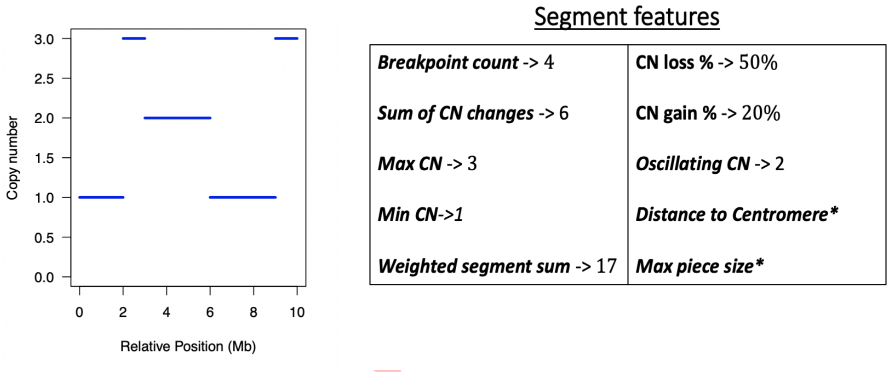
Figure 5: *Properties about the segments when the size is changed in a range between 5Mb and 25Mb.*

To help us decide the optimal size, we created a collection of plots that can be seen in Figure 5. These plots contain information about different properties of the segments when the size is changed in a range between 5Mb and 25Mb. The first plot shows the mean number of breakpoints per segment,. We do not need to have a breakpoint in each segment, but a very low number of breakpoints shows that there are many monotone segments consisting of only one piece, so the highest this value is, the better. The second plot represents the number of segments per sample, the highest this number is, the better quality we will get in the results. Moving on, we have the total information lost per sample, this is the propertie that was mentioned above, and calculates the amount of information that will not be collected in each sample. We are interesting in collecting as much information as possible, so we will try to keep this number to the minimum. Finally, we have a plot showing the ratio of unique divided by total segments. To make this plot we made use of some of the features that will be explained in the next section. The aim of this plot is to show an approximation of the number of segments that will have unique properties compared to the total number of segments (it is an approximation because we have not chosen the categories for the features yet). We will be interested in having a low ratio for the algorithm to perform better.

As we have just seen there are some contradictions in the properties, we are interested in having a high number of breakpoints per segment, while at the same time not loosing much information and also having enough segments per sample. The optimal number of 10Mb was chosen as a midpoint that allows us to collect most of the information from the samples, while at the same time having a large amount of interesting segments (most of them with over 1 breakpoint).

2.4 Selecting the features

Moving on, the next decision is the features to study. We can get ideas from other papers like [2] or [7]. Also, since our idea is to compare the results to the ones obtained by [2], we are interested in having similar features to study. Remember that, for the moment, we are only considering the first alternative, so the segments are defined as 10Mb pieces in the genome. This alternative allow us to explore a higher number of features compare to the other definition of segments. Inside a 10Mb segment there may be multiple pieces of DNA with different CN values as we can see in the left picture of Figure 4.



Segment features

Breakpoint count -> 4	**CN loss %** -> 50%
Sum of CN changes -> 6	**CN gain %** -> 20%
Max CN -> 3	***Oscillating CN*** -> 2
Min CN->1	***Distance to Centromere****
Weighted segment sum -> 17	***Max piece size****

Figure 6: *Representation of an example 10Mb segment with the description of the features to study.*

Moreover, once we have divided each genome into segments, we will only consider those segments with at least one copy number aberration. This implies that samples will have more segments than others. The reason to do this is that otherwise there would be many "normal" segments without any copy number aberrations and the algorithm would detect them as something important to include in the signatures. By deleting this segments we only keep the distinctive information for each sample.

There are many possible features to study, but some of them may be correlated and give us similar information. To avoid adding extra noise, we will choose only a few of these features that represent the most important properties of each segment. Initially, we start with the features that are shown in Figure 6. This figure represents an example of a 10Mb segment and shows the values for all the initially proposed features. The *Breakpoint count* represent the number of places where there is a change in the CN value inside the 10Mb segment. Following the same idea, the *Sum of CN changes*, sums the absolute value of the differences in all the CN breakpoint positions. The next two features are *Max CN* and *Min CN*, and they represent the Maximum and Minimum copy number value found in the segment. Then we have the *Weighted segment sum*, which calculates the product of the size of the pieces with same CN, multiplied by the number of copies. That is:

$$\text{Weighted Segment Sum} = \sum_{p\ in\ pieces} \{\text{Copy number of } p\} \times \{\text{Size of } p\}$$

In the example from Figure 6 this would be equivalent to $1 \times 2 + 3 \times 1 + 3 \times 2 + 3 \times 1 + 1 \times 3 = 17$. Moving on, the *CN loss %* and the *CN gain %* display the proportion of DNA in the 10Mb segment that have copy number losses or gains respectively. In Figure 6, the first and fourth pieces have copy number losses and have a combined size of 5Mb, representing a 50% of the 10Mb genome, and a similar scenario occurs for the CN gains. The next feature is the *Oscillating CN* which looks for loops in the CN values inside the segment. To have an oscillation given a piece of DNA with a certain CN value, we need to have at least one other piece with a different CN value and eventually a new piece with the initial CN. In the example from above, the first piece has $CN = 1$, then there are two pieces with $CN = 3$ and $CN = 2$, and then a piece with the same number of copies as the first one. This represents an oscillation between the first and fourth pieces of the segment.

The last two features in the table from Figure 6 are the *Distance to Centromere* and the *Max piece size*. The first one, as the name indicates, represents the distance to the centromere and the second one collects information about the size of the longest piece inside the segment. Both of these features are potentially important. However the definition of the segments as in Alternative 1 limits the quality of the information. Thus, these features will be used when we study the Alternative 2 for the definition of segments.

Once all the features have been defined, it is time to decide which are the optimal ones to represent the main characteristics of the segment. In order to do that, let us start by looking at the correlations between the features. As we can see in Figure 7, there are several cases of high correlation between features. This plot is informative, but it is only one of multiple factors to consider when it comes to choosing the features.

The first four features of the plot are highly correlated, either positevely or negatively. The *Weighted Segment Sum* (WSS) will be selected to represent these four features, as it allows more variability between the possible values and it will be more informative when it comes to understanding the meaning of the signatures.

Moving on, the last two features in the plot also show high correlation. However they represent different concepts and the high correlation can be explained by the fact that a higher number of breakpoints increases the probability of having an oscillation, even if it is by chance. Thus, we will keep both of these

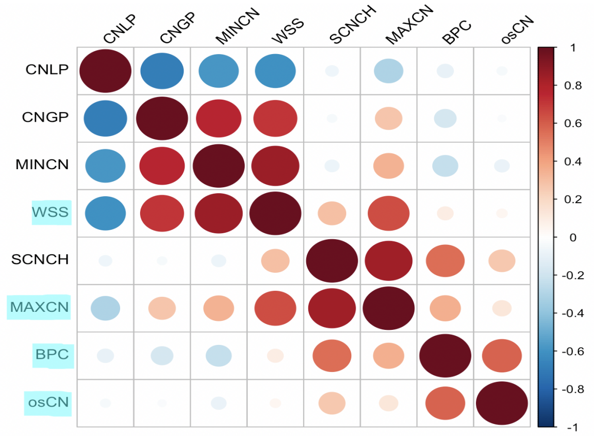


Figure 7: *Correlation heatmap between the features of the segments. The bigger the size of the circle, the higher level of correlation. Blue circles indicate negative correlation and red circles indicate positive ones. The chosen features are highlighted in light blue.*

features for the model.

Lastly, the *Sum of CN change points* (SCNCH) and *Max CN*show a high correlation as well. This is because whenever there is an unusually large CN value in a segment, the neighbor values tend to be much smaller, so there will also be a large jump between the pieces. Out of these two, the *Max CN* was the final choice for the model as its combination with the *Breakpoint count* (BPC) will give similar information to the *Sum of CN change points*.

As a summary, the final features are the ones highltighted in lightblue in Figure 7, the *Weighted Segment Sum* (WSS), *Max CN*, *Breakpoint Count* and *Oscillating CN* (osCN). The *Weighted Segment Sum* will capture the general information about the segment. The *Max CN* will look for unusual peaks and regions with a high number of copies. Moreover, the *Breakpoint count* will tell us the number of pieces of the segment, and thus its variability. And finally the *Oscillating CN* will look for segments with this particular phenomena, which was also studied in [2].

2.5 Dividing the features into components

The next step in the process is categorizing the features. The algorithm requires to have only a few categories per feature as it will count the repetition of each combination of features during the process to find relations afterwards. This means that our variables will not only need to be categorical, but will also need to have few categories (less than 10 for our amount of data).

There are several possible approaches to divide a feature into categories. In the case of the *Breakpoint count* or the *Max CN*, the features are already divided into categories, and most segments (over 90%) can be assigned to $BPC < 5$ and $Max_CN < 6$. This incites us to just create a category for $BPC \geq 5$ and $Max_CN \geq 6$, and then keep the other values as individual categories. This is a very simple approach which can only be used with these features because of the way the data is distributed.

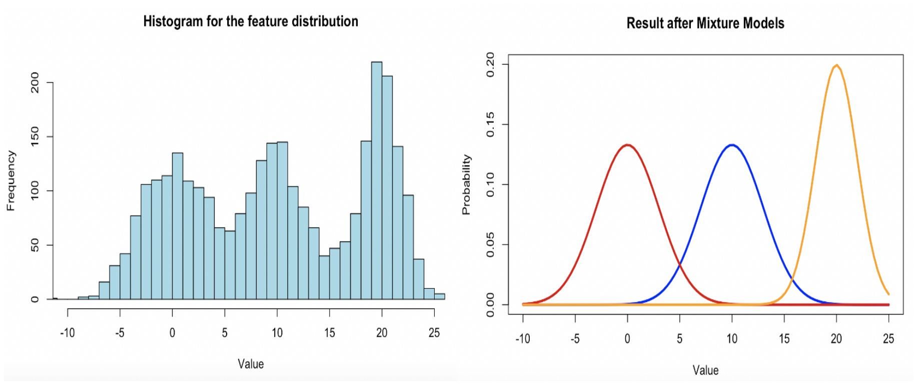


Figure 8: *Example of the use of mixture models. The feature distribution is divided into 3 normal distributions that would be the components for that feature.*

For the case of the *Oscillating CN*, we have the problem that in most of the segments we will not have this event, and for the ones where it happens the number of oscillations will usually be just one. In [2], they studied the length of oscillating CN, which only makes sense when the features are analyse in a sample case. For the segment scenario of our work it is already hard enough to find an oscillation event. Thus, the final decision was to make this feature binary, having $osCN = 1$ if there is at least one oscillating CN event and $osCN = 0$ if there are no oscillations.

Finally we have the *Weighted Segment Sum*, which is more complex than the previous features. In this case, the number of possible values is much larger, so we cannot repeat the same procedure as with the first features. A first alternative would be to use mixture models, which was used in [2] to categorize the sample features. An example of mixture models can be seen in Figure 8

Given the distribution of counts for a feature, this algorithm divides the distribution into combinations of Normal or Poisson[1] (depending on the feature) distributions. The idea behind this is that the values for a feature are assumed to come from different distributions, each representing a component of the feature. Once the main distribution for a feature has been divided into different Normal or Poisson distributions, given a new value, the density functions for these distributions will be used to decide the probability that the new value belongs to each of the components.

This procedure is based on statistical approximations and the results will be accurate in the sense that the combination of the components will be the feature distribution. However, the meaning of the components may be confusing in some scenarios and even if the approximations are accurate, they may not be useful when it comes to understanding the signatures. This is why, although the first approach to categorize this feature was tu use mixture models, the final

[1] The mixture models can be used with other distributions as well, but for the purpose of our work we just considered these two.

decision was a much simpler procedure. The values for WSS were calculated for the case where there is a segment consisting of a singular piece with $CN = 1, 2, 3, 4$ or 5. For the case that we are studying of a 10Mb segment, these values are $1e6, 2e6, 3e6, 4e6$ and $5e6$. These were used as the dividing points for the components, so the first component would be formed by the segments whose WSS lies between 0 and $1e6$, the second one by the ones whose WSS is between $1e6$ and $2e6$... Up until the final component for segments with WSS higher than $5e6$.

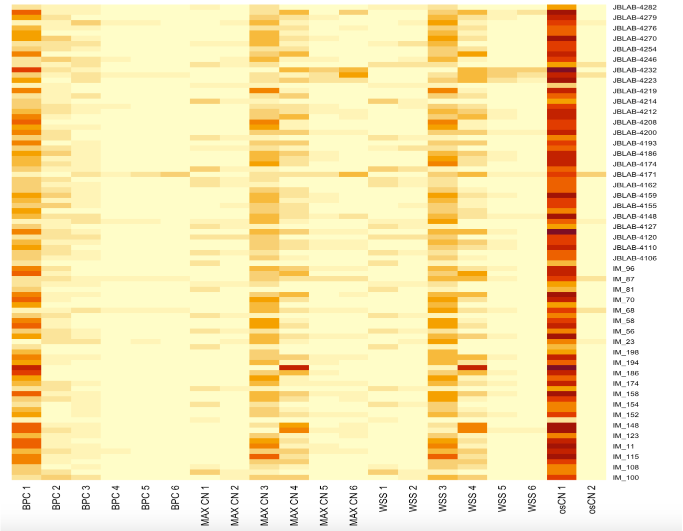


Figure 9: *Heatmap for the sample by component matrix.*

The final components can be seen in Figure 9. The rows indicate the sample names and the columns denote the components. Each square represents the number of segments with that particular component in the indicated sample. The color of the squares is ranked from light yellow, which indicates 0 segments to dark red for the maximum number of segments. Each segment belongs to exactly one component of each feature, that is why the feature $osCN$ (oscillating CN), which has only two components, has the highest number of segments. Also remember that only those segments with CN aberrations are considered. This is why there are some rows with mostly light yellow colours, indicating that the number of segments with copy number aberrations in that sample is low.

2.6 Selecting the number of signatures

Once the features have been categorized we are almost ready to run the algorithm. The only parameter we need to decide is the number of signatures that will be considered. This is again a decisive step that will affect the final result. There have been different approaches to choose the optimal number of signatures. Some of them are based on statistical information criteria methods, such as AIC or BIC. The problem with these criteria is that small deviations between the specified probabilistic models and the reality sometimes lead to additional mutation signatures being selected to compensate for those deviations.

Shiraishi gives an alternative procedure. For a range of possible number of signatures, we will calculate the likelihood and standard deviation of the estimated parameters. Then, for each of the possible signatures, we look at the correlation for the exposures of the signatures. If this correlation is high, then there will be at least two signatures which are very similar and we will try to avoid this. The final optimal number of signatures is chosen as the one with the best combination of high likelihood, low standard deviation of the parameters and low correlation between signature exposures.

The authors of [3] admit that this method is not optimal as it requires a manual inspection and a subjective choice. Nevertheless the optimal number of signatures also depends on our interests. For example, we may be willing to have two correlated signatures if they differ in a key feature of interest. Thus, having a manual procedure to select the optimal number of signatures may not be as incoherent.

On the other hand, [2] uses a series of measures from the NMF package to determine the optimal number of signatures. These include the cophenetic, dispersion, silhouette and sparseness coefficients. The cophenetic coefficient is a way to measure the correlation between the signature exposures. Then, the dispersion coefficient calculates the stability of the results from the NMF algorithm. To measure the distance between the signature definitions we use the silhouette coefficient, the higher this coefficient is, the closer the signatures are. Lastly we calculate the sparseness of the columns from the resulting matrices of the NMF algorithm. This is measured using a real number between 0 and 1 that is equal to 1, if and only if the column contains a single nonzero component, and is equal to 0 if and only if all components of the column are equal.

These coefficients are compared from a range of signatures using on one hand the original data of the sample by component matrix and, on the other hand, some randomized matrix of the same data. The final choice was 7, as it was the highest number where the sparseness of the signature by component matrix obtained with the original data was lower than the spareness obtained with a random initial matrix.

As we can see, both of these procedures have a final step where we have to make a decision. This decision is not straightforward and will depend on our interests. Having a higher number of signatures may allow us to have more precise exposures of the components. However, as we increase the number of signatures, they will be more correlated and we may be giving importance to the noise of the data.

The final decision for our work was to combine some of the criteria from both [2] and [3]. We will focus on the correlations between the definitions of the signatures, the sparseness of the signature by component data and also the log-likelihood of the results from the algorithm. All of this can be seen in Figure 10. To create these plots, we run the algorithm for 100 iterations for a range of signatures between 3 and 12. The mean correlation between signature definitions reaches a local minima at 8 signatures. Moreover the number of highly correlated signature pairs is only one for 8 signatures (the only time it is 0 is for 3 signatures), and it increases when we choose a higher number than 8. The sparseness is not the maximum at 8 but the difference with the maximum

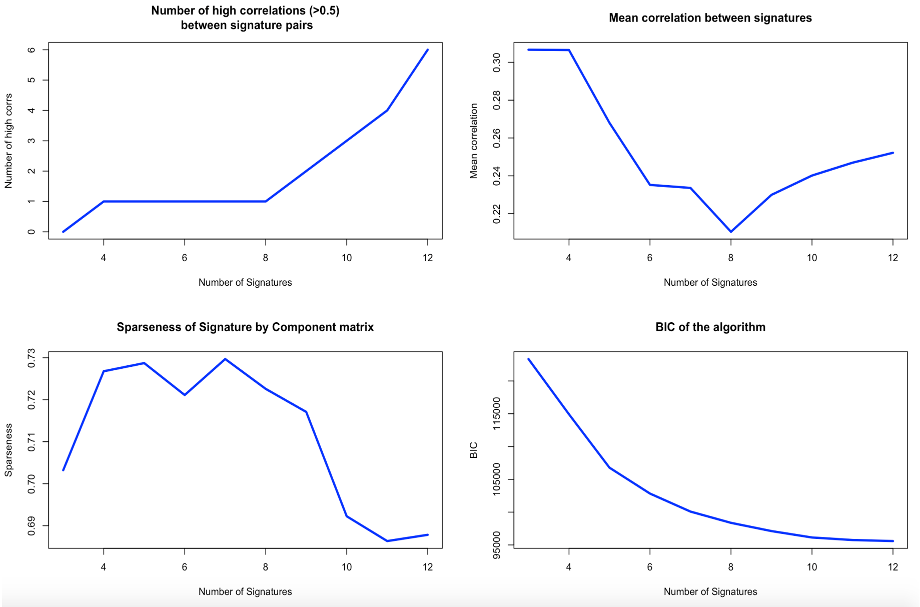


Figure 10: *Plots showing a variety of properties about the signature by component matrix and the performance of the algorithm as we vary the number of signatures from 3 to 12.*

value is around 0.01, which is very low. Finally the BIC is strictly decreasing, so it does not provide any additional information. Thus, the optimal number of signatures was chosen to be 8.

3 Results

3.1 First results for the initial choices

Now that the methods have been explained and the main decisions have been made we are ready to run the algorithm and obtain some initial results. As a summary, the genome for each sample is divided into 10Mb segments and only those with CN aberrations are selected. Then, for each segment, we calculate the *Breakpoint Count* (BPC), *Maximum CN* (Max CN), *Weighted Segment Sum* (WSS) and *Oscillating CN* (osCN) for each segments. These features are divided into components, leaving us with a total of $6 + 6 + 6 + 2 = 20$ components to study. Moreover, the final decision for the number of signatures was 8.

One of the main goals is to define the signatures and describe the features of each sample as a combination of the signatures. With this idea in mind, the first result that we will introduce will be the sample by signature matrix. This is the main way to describe the signature exposures for each sample. It describes the proportion of each signature that is present in each sample.

A heatmap for the sample by signature matrix can be seen in Figure 11. The columns represent the signatures and the rows the samples. Each colored cell represents the expression level for a signature in a particular sample. The colors

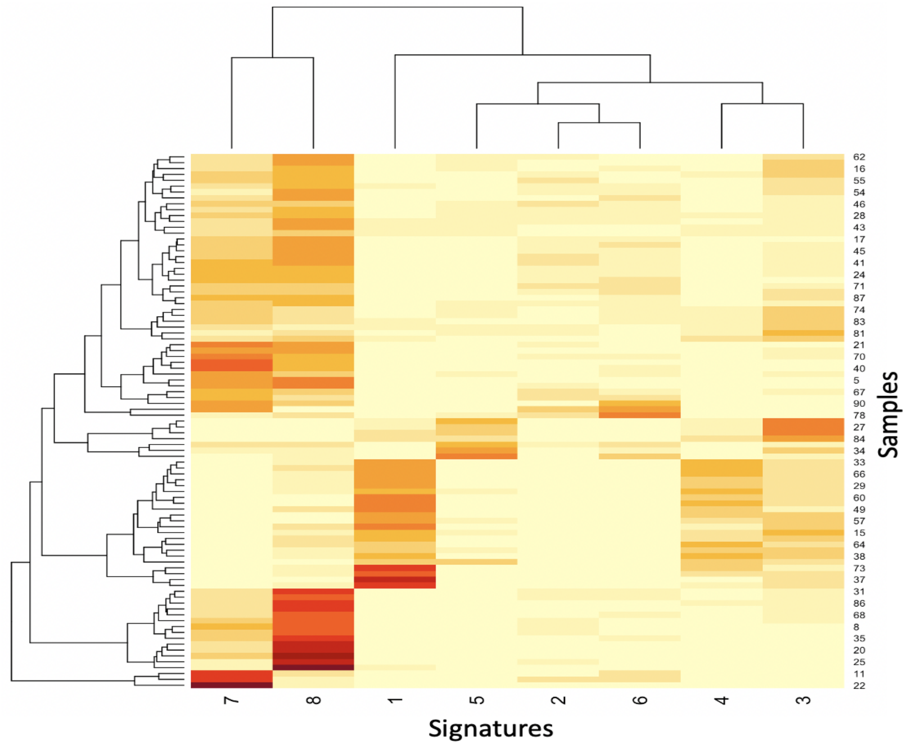


Figure 11: *Heatmap for the sample by signature matrix. The dendograms on top and on the righthand side show the relationships between signatures and samples respectively.*

range from a light yellow for the lowest level of expression to a dark red for the highest level.

As we can notice, both the columns and the rows are reorganized so that the most similar samples and signatures are together. On the top and on the left hand side of the main heatmap we can see some trees representing these relationships.

There are a variety of exposures for the signatures across the samples. Some signatures, like 7 or 8, are present with high levels in many samples, while other signatures, such as 2 or 6, are less frequent. However, this does not mean that the signatures are less important. In fact, since they only have high levels in a few samples the presence of those signatures becomes a distinctive characteristic for the samples were they are expressed.

To fully understand the meaning of the sample by signature matrix, we need the signature by component information. The heatmap for this matrix can be seen in Figure 12. The colors work the same way as in Figure 11, light yellow show the lowest levels for that component and the darker it gets, the highest presence of that component in the corresponding signature.

For example, we saw before that signature 8 was highly expressed in many

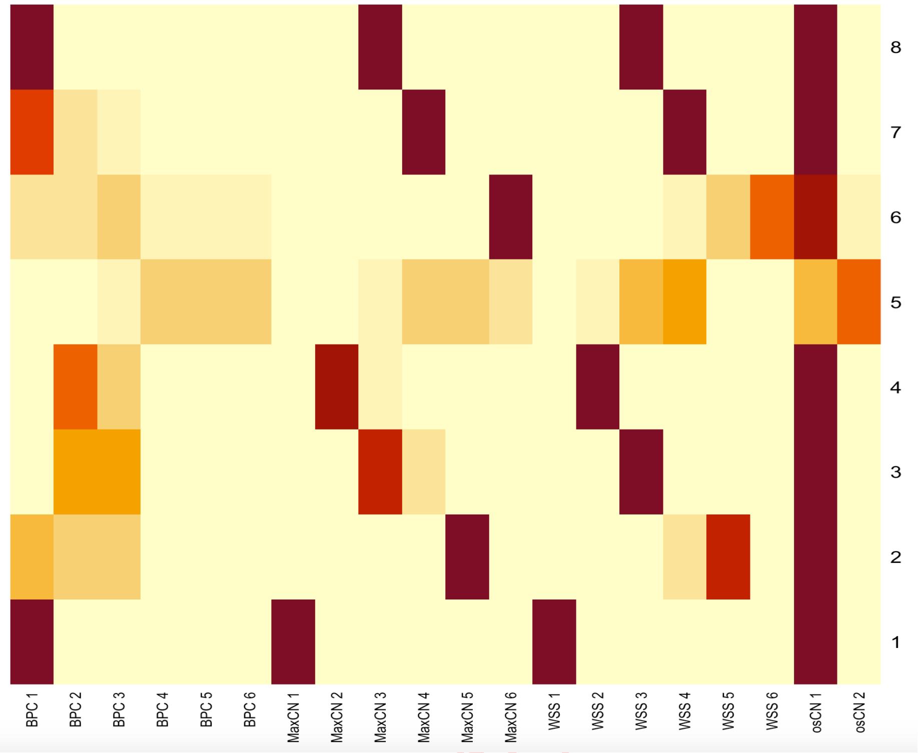


Figure 12: *Heatmap for the signature by component matrix.*

samples. Now, in Figure 12 we can see that this particular signature is characterized by low number of Breakpoints, with a maximum copy number of 3, an average copy number of also around 3 (WSS=3) and also none oscillations. Thus, the samples with high levels of presence for this signature will likely be characterized by long segments with 3 copies.

4 Appendix 1: NMF and Mixture Model details

As was explained above, once the features are selected and the distributions of each feature accross samples are computed, mixture models are applied. For a given feature, we will assume that the distribution is formed by the combination of multiple sub-distributions coming from different clusters. For this experiment it was assumed that the sub-distributions were either Poisson (when the feature involved counts) or Normal (otherwise), but other distributions could have been used. They used the FlexMix V2 package in R [6] to apply the algorithm and obtain the sub-distributions for each feature. Each of these sub-distributions is defined as a component of the feature and this allows us to reduce the dimensionality and the optimal number of components was selected as the one with the lowest Bayesian information criterion (BIC), resulting in a total of 36 components, as can be seen in Figure 1. So for each CNA event, we can calculate the posterior probability that this event comes from each of the components (or sub-distributions). By doing this for all CNAs and summing the posterior probabilities we can create the sample by component matrix.

Lastly, the Brunet specification [4] of the NMF algorithm was used to decompose the sample by component matrix. This step is the one that will be replaced by the methods from [3], so let us explain how it works in more detail. The starting point is the sample by component matrix, that contains, for each sample, the expression levels of each of the components in that sample.

As it is explained in [4], our goal is to find signatures, each defined as a positive linear combination of the 36 components. We can then approximate the component's expression pattern of samples as positive linear combinations of these signatures.

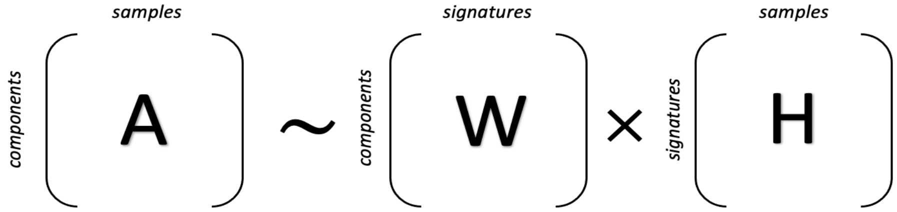


Figure 13: *Matrix summary of Brunet's NMF algorithm.*

Mathematically, this corresponds to factoring matrix A into two matrices with positive entries, $A \sim W \times H$. Matrix W has size $N \times k$, with each of the k columns defining a signature; entry w_{ij} is the coefficient of component i in signature j. Matrix H has size $k \times M$, with each of the M columns representing the signature expression pattern of the corresponding sample; entry h_{ij} represents the expression level of signature i in sample j.Given a factorization $A \sim W \times H$, we can use matrix H to group the M samples into k clusters. Each sample is placed into a cluster corresponding to the most highly expressed signature in the sample; that is, sample j is placed in cluster i if the h_{ij} is the largest entry in column j. Given a positive matrix A of size $N \times M$ and a desired rank k, the NMF algorithm iteratively computes an approximation $A \sim W \times H$, where W and H are non-negative matrices with respective sizes $N \times k$ and $k \times M$. The method starts by randomly initializing matrices W and H, which are iteratively updated to minimize a divergence functional. The functional is related to the Poisson likelihood of generating A from W and H,

$$D = \sum_{i,j} A_{i,j} \log \left(\frac{A_{i,j}}{(WH)_{i,j}} \right) - A_{i,j} + (WH)_{i,j}.$$

At each step, W and H are updated by using the coupled divergence equations :

$$\begin{aligned} H_{au} &\leftarrow H_{au} \frac{\sum_i W_{iu} A_{iu}/(WH)_{iu}}{\sum_k W_{a,k}} \\ W_{ia} &\leftarrow W_{ia} \frac{\sum_u H_{au} A_{iu}/(WH)_{iu}}{\sum_v H_{a,v}}. \end{aligned}$$

The article [4] also contains some indications about how to choose the number of signatures. The are already a part of the NMF package for R.

References

[1] Alexandrov, L., Nik-Zainal, S., Wedge, D. *et al.*; Signatures of mutational processes in human cancer. Nature **500**, 415–421 (2013).

[2] Macintyre, G., Goranova, T.E., De Silva, D. *et al.*: Copy number signatures and mutational processes in ovarian carcinoma. Nat Genet **50**, 1262–1270 (2018).

[3] Shiraishi Y, Tremmel G, Miyano S, Stephens M: A Simple Model-Based Approach to Inferring and Visualizing Cancer Mutation Signatures. PLOS Genetics 11(12): e1005657, (2015) .

[4] Brunet, J.-P., Tamayo, P., Golub, T. R. and Mesirov, J. P. Metagenes and molecular pattern discovery using matrix factorization. Proc. Natl Acad. Sci. USA 101, 4164–4169 (2004).

[5] NMF package in *R*

[6] FlexMix V2 package in R

[7] Lisui Bao, Xiaoming Zhong, Yang Yang, Lixing Yang. Mutational signatures of complex genomic rearrangements in human cancer (2021)