Università degli studi di Torino

**SCUOLA DI SCIENZE DELLA NATURA**

**Corso di Laurea Magistrale in Fisica dei Sistemi Complessi**

**Master degree thesis**

# A topic model approach reveals hidden structures in datasets of healthy and cancer tissues.

**Supervisor:**
**Prof. Michele Caselle**

**Co-supervisor:**
**Dott. Matteo Osella**

**Author:**
**Filippo Valle**

**Examiner:**
**Dott. Matteo Cereda**

**Academic year 2018/2019**

The imagination of nature is far, far greater
than the imagination of man[1].

Richard P. Feynman.

[1] What do you care what other people think?

# Abstract

The interest in studying complex systems is increasingly spreading. Complex systems can be found anywhere and many common behaviours are observable among them, systems with different origins and purposes may share some statistical laws.

An example can be the Zipf's law, well-known in linguistics and texts analysis. It can be easily observed in the distribution of gene expression in different samples of cancer tissues.

In recent years, datasets with a large amount of cancer samples' data are available, the most complete is The Cancer Genome Atlas (TCGA). From this dataset, it is easy to obtain, for example, gene expression data from RNA-sequencing experiments together with a lot of metadata about the samples themselves. Another dataset containing healthy tissues (GTEx) will be analysed for comparison and benchmark.

If one studies the number of samples in which a gene is expressed above a certain threshold, the so-called occurrence, it is easily verified that there are different kinds of genes. Some are present in the majority of samples, some others are present only in a subset of the whole dataset. The same behaviour can be found analysing words in a corpus of texts; some words, such as *the*, are present everywhere, other specific words are present only in texts regarding a certain subject. This suggests that there are similarities between a system of words and documents and a system of genes and samples.

Given a corpus of documents, these can be classified by their specific subject. Similarly, samples can be classified, for example, by the tissue they come from or by the type of the disease they are referred to.

The similarities between gene expression and linguistics data suggest the possibility to use topic modelling to classify data and separate samples and genes in different clusters. Topic modelling is a set of clustering algorithms in networks' theory. Given a set of words and documents, these algorithms describe documents as a mixture of topics. Topics are nothing but communities of words each one with a given probability.

Purpose of this work is to build a bipartite network of genes and samples and use topic modelling to find communities. The goal is to separate samples depending on the site the tumour was and the disease type of the sample. Moreover, it is possible to separate genes depending on their specific functions. Once a community structure of genes emerges, it is possible to run a hypergeometric test on the whole set to verify if it reveals some type of enrichment and to inspect common properties among genes.

The specific algorithm used in this work is particularly unique because it needs no priors and makes no assumptions on the data. Moreover, it can be set to accept overlapping clusters so it is possible to find genes belonging to different topics and it

can be hierarchic. All these facts empower a lot of new possibilities to investigate a network.

A hierarchic approach makes it possible to classify data at different layers. An ideal goal would be to separate healthy and diseased samples at the first layer, then separate them by tissue, then by the tumour type and so on.

# Contents

# Chapter 1

# Introduction

In recent years the study of complex systems is becoming more interesting especially when some different systems that share some fundamental properties are found. Network theory has been proven to be a useful proxy to model and represent such complex systems.

This work wants to study and find universal statistical laws in different kinds of biological systems. If one finds that two different systems share some statistical laws and that they have a somehow similar data structure, therefore it is possible to use tools developed from different fields to gain more information. In particular, two datasets containing information about some human healthy and diseased tissues will be analysed. These data come from biological experiments of RNA-sequencing.

The ultimate goal of this work would be to study, develop and build a machine learning's model which is able to classify cancer tissues and gain information from healthy tissues as well. Separate cancer types and ultimately sub-types is not always easy clinically and that's why the interest in developing a method able to well classify this kind of data is increasing [1].

The methods to gain this goal are firstly derived from linguistics; in particular, a topic model approach will be widely described. A hierarchic approach will be useful to gain different layers of information.

In chapter 2, I will describe the datasets used and introduce some basic biological properties of these datasets. In particular, I'll use two datasets of gene expression data from cancer and healthy tissues.

In chapter 3, I will describe the basics of component systems and give some useful mathematical definitions. Here I will show that RNA-sequencing data have many aspects in common with linguistics data. Examples of Zipf's law, well-known and in-depth studied in linguistics, will demonstrate that different sources of data (genomic and linguistics) can share some statistical properties. Some analyses will be shown to explain the different behaviour of different tissues.

In chapter 4, I will study the gene expression across samples of all the genes. This analysis is preparatory to the following sections where some gene selection will be necessary.

Demonstrated that linguistics and biological data share some statistical laws, in chapter 5, the core of this work, I will describe how topic modelling can perform network analysis on these datasets. Topic modelling is an advanced clustering algorithm developed in linguistics to classify text and used in different fields of science. Different approaches to topic modelling are possible starting from the standard ones [2] to some new proposals [3, 4, 5]. Using topic modelling one would find the inner structure of the data. One would find clusters such that all samples in a cluster share the tissue or the tumour type. Benchmarks and metrics to test and evaluate this algorithm will be widely discussed.

In chapter 6, I will sum up the results and propose some future developments of this work.

Many methods of the pipeline, written in C++ using openMP and Boost [6], are encapsulated in a tool available at `https://github.com/fvalle1/tacos`. During this work, I used different python libraries such as pandas [7], scipy [8], numpy [9] and matplotlib [10]. Some advanced analysis required Tensorflow [11] and pySpark [12]. The topic modelling stochastic block model's minimization functions are implemented in the graph-tool library [13]. Computing resources were made available by EGI Foundation [14] and from C$^3$S [15].

The full work repository is available on GitHub© at `https://github.com/fvalle1/master_thesis`. The full pipeline is runnable inside a Docker© container that can be pulled from `https://hub.docker.com/r/fvalle01/thesis`.

# Chapter 2

# Datasets presentation

## 2.1 Data from RNA-sequencing experiments

Data considered in this work come from RNA-sequencing [16] experiments. These experiments aim to quantify how much a gene is expressed in a particular sample of a given tissue. RNA-Sequencing data provide a unique snapshot of the transcriptomic status of the sample. Data considered are from bulk experiments, this means that each value is an average over multiple cell.

Briefly, long RNAs are first converted into a library of complementary DNA (cDNA) fragments through either RNA fragmentation or DNA fragmentation. Sequencing adaptors are subsequently added to each cDNA fragment and a short sequence is obtained from each cDNA using high-throughput sequencing technology. The resulting sequence reads are aligned with the reference genome or transcriptome, and classified as three types: exonic reads, junction reads and poly(A) end-reads. These three types are used to generate a base-resolution expression profile for each gene.

The general steps to prepare a cDNA library for sequencing are, in general:

- RNA Isolation: RNA is isolated from tissue and the amount of genomic DNA is reduced;

- RNA selection/depletion: to analyse signals of interest, the isolated RNA can either be kept as is, or filtered for RNA that binds specific sequences. The non-coding RNA is removed because it represents over 90% of the RNA in a cell, which, if kept, would drown out other data in the transcriptome;

- cDNA synthesis: RNA is reverse transcribed to cDNA (DNA sequencing technology is more mature). Fragmentation and size selection are performed to purify sequences that are the appropriate length for the sequencing machine. Fragmentation is followed by size selection when either small sequences are removed or a tight range of sequence lengths are selected. Because small RNAs like miRNAs are lost, these are analysed independently. The cDNA for each experiment can be indexed with a hexamer or octamer barcode, so that these experiments can be pooled into a single lane for multiplexed sequencing.

To collect gene expression data is sufficient to count how many reads are mapped to a specific exon or gene. The ultimate output of this analysis, where this work begins, are nothing but lists of gene expression values for each sample.

**Different normalization are available**
Usually gene expression data can be normalized in different ways, for example, it is possible to use:

- counts;
- RPK;
- TPM;
- FPKM.

Count reads correspond to raw data. Counts need no normalization to be treated but may be biased. For example, longer genes may have more reads than shorter ones just because they are longer. This is why other kind of normalization can be used. Note that this is not always the case: in fact, some sequencing techniques consider just the start of the gene, so the gene length doesn't matter.

RPK[1] normalization removes the length bias dividing counts by the gene length $L$,

$$\text{RPK} = \frac{\text{counts}}{L}$$

. This solves some problems but doesn't take care of the different sizes of the transcript in different samples.

FPKM[2] calculation normalizes read counts dividing them by the gene length and by the total number of reads mapped to protein-coding genes in that sample,

$$\text{FPKM} = \frac{RC_g * 10^9}{RC_{pc} * L}$$

being

- $RC_g$: Number of reads mapped to the gene;
- $RC_{pc}$: Number of reads mapped to all protein-coding genes;
- $L$: Length of the gene in base pairs.

When dealing with FPKM it is necessary to put some thresholds: in particular FPKM below 0.1 and above $10^5$ should not be considered, maybe these values come from some kind of experimental error.

TPM[3] tries to unify the sizes of the samples: TPM $= \frac{RC_g * 10^9}{\sum_{g'} \left( \frac{RC_{g'}}{l_{g'}} \right) RC_{pc} * L}$.

[1] Reads Per Kilobase of transcript

[2] Fragments Per Kilobase of transcript per Million mapped reads

[3] Transcript Per Million

In this work the idea is to not introduce any normalization, so when possible raw counts will be considered. Sometimes, especially if it is necessary to compare different sources, TPM or FPKM will be taken in account. Some analysis, as the distribution of the sample size, need to be done without TPM, because the quantity studied is the one the normalization trashes out.

## 2.2 Datasets version

In this work two datasets were used. The first one contains RNA-sequencing data of post-mortem collected samples. It is the Gene Tissue Expression (GTEx) dataset [17]. GTEx *2016-01-15 v7 RNASeQCv1.1.8* version was downloaded[4]. GTEx contains 11688 samples of 53 tissues. For many of them, a sub-tissue label is available. As highlighted in [18] these data present a challenge to clustering tools, because of both the relatively large number of samples and the complex structure created by the inclusion of many tissues.

The other dataset considered is The Cancer Genome Atlas (TCGA) [19]. Data were collected via Genomic Data Commons tools[5] considering *Gene Expression Quantification* as data type, *Transcriptome Profiling* as data category, *RNA-Seq* as experimental strategy, *HTSeq - Counts* or *HTSeq - FPKM* as workflow type. On TCGA there are 12683 samples and 68 primary sites or tissues. On this dataset there is a great quantity of metadata, in particular the *disease type* will be considered in this work.

The third source of data considered in this work is [20]. Its authors tried to unify GTEx and TCGA [21], when possible.

#### Protein-coding gene selection

Each dataset contains information on approximately 60000 elements with a different *ENSG* identifier. Only $\simeq$ 20000 of this are protein-coding genes, using Ensemble[6] protein-coding genes are selected. In chapters 3 and 4 I will propose some analyses that explain the different behaviour of coding and non-coding genes.

[4] https://gtexportal.org/home/datasets

[5] https://portal.gdc.cancer.gov

[6] https://ensemble.org

# Chapter 3

# Data structure

The data studied in this work can be represented as component systems. These component systems can be represented by a two-dimensional matrix in which rows represent components and columns are the possible realizations buildable given a subset of the components. Entries of this matrix are the number of the components on the row needed during the realization of the column. In figure 3.1 an example of this kind of matrices.

$$\begin{array}{cc} & \text{Realizations} \\ \text{Components} & \begin{pmatrix} n_{11} & n_{12} & n_{13} & \cdots & n_{1R} \\ n_{21} & n_{22} & n_{23} & \cdots & n_{2R} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n_{N1} & n_{N2} & n_{N3} & \cdots & n_{NR} \end{pmatrix} \end{array}$$

Figure 3.1: Matrix representing component systems with $i = 0 \dots N$ rows and $j = 0 \dots R$ columns. The entry $n_{ij}$ represents the abundance of component $i$ in realization $j$.

## 3.1 Component systems

The most common example of such systems is a set of books. In this case one puts on the rows the words in the whole vocabulary and the books' titles on the columns. The entry that corresponds to row $i$ and column $j$ is the number of times the word $i$ appears in the book $j$. The same happens if one considers Wikipedia's pages. Other examples are: Lego® sets where components are the Lego® bricks and realizations the Lego® packages, and protein domains; all these were described and well studied in [22, 23].

Given a matrix with $N$ components on the rows and $R$ realizations on the columns and relative abundances $n_{ij}$ as the entries, it is interesting to study some quantities that are universal and general characteristics of component systems.

First of all, the **occurrence** of a component is defined as

$$O_i = \frac{\sum_{j=1}^{R}(1 - \delta_{n_{ij},0})}{R}. \tag{3.1}$$

It is the fraction of realizations in which the component's abundance is not null. A component that is present in all the realizations has got $O_i = 1$, the ensemble of the components with $O_i = 1$ is known as the **core**. Components with high ($\simeq 1$) occurrence are present in mostly all realizations of the datasets. In linguistics articles, such as *the*, are present everywhere, so they have high occurrence. Components with low occurrence ($\simeq 0$) are present only in a few realizations and are the most specific ones [24].

The sum across all columns, or the number of times component $i$ appear in the dataset, is called **abundance** of the component and is defined as

$$a_i = \sum_{j=1}^{R} n_{ij}; \tag{3.2}$$

dividing this by the global abundance, or the total number of components in the dataset,

$$a = \sum_{i=1}^{N} a_i \tag{3.3}$$

naturally brings to the **frequency of a component** in the whole corpus

$$f_i = \frac{a_i}{\sum_{c=1}^{N} a_c}. \tag{3.4}$$

The abundance of a component divided by the sum of all the abundances in a realization gives the **frequency** of the component in that specific realization

$$f_{ij} = \frac{n_{ij}}{\sum_{c=1}^{N} n_{cj}}. \tag{3.5}$$

The sum of all abundances in a realization,

$$M_j = \sum_{c=1}^{N} n_{cj} \tag{3.6}$$

represents the **size** of the realization. In gene expression this is the size of the transcript.

It is expected that frequencies distribute according to the so-called Zipf's law

$$f_i \propto r_i^{-\alpha} \tag{3.7}$$

where $r$ is the rank: the position of a component when sorting, in descending order, all components by their frequencies in the whole dataset.

## 3.2 Universal laws in gene expression datasets

### Cancer samples: TCGA

The first interesting quantity, analysed in the TCGA, is the sorted abundance. This gives the possibility to approach the so-called Zipf's law. In figure 3.2, it is shown the

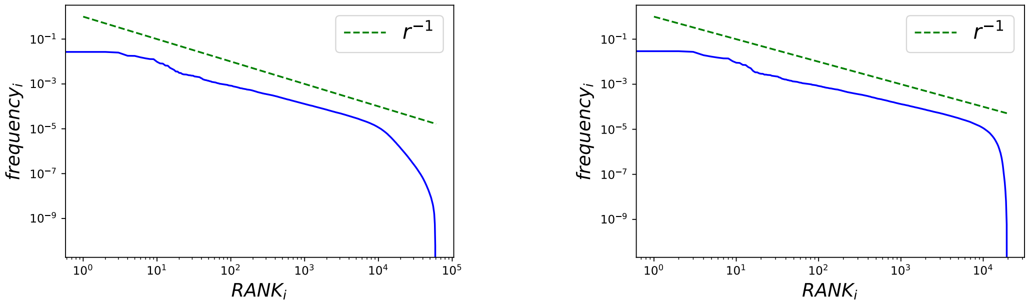


Figure 3.2: Zipf's law from FPKM normalized data. On the right considering only protein-coding genes.

frequency rank plot. Interestingly, this kind of data distributes according to a power law, this same behaviour can be found in many systems such as linguistics' ones [24]. Another interesting fact is that the analysis of both coding and non-coding genes gives a double-scaled power law. This happens since non-coding genes are specific and rare, so their frequencies are quite small compared to protein-coding genes' ones.

Changing the normalization and considering counts instead of FPKM, the result is quite similar. The power law is flatter, meaning that genes have more similar abundances in the whole dataset.

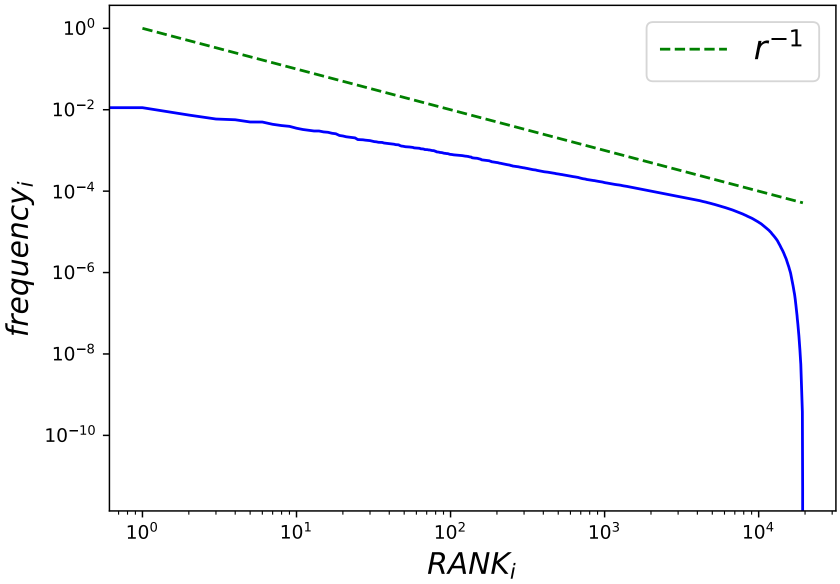


Figure 3.3: Zipf's law of protein-coding genes considering counts.

**Healthy tissues: GTEx**

The same analysis can be performed on healthy samples from GTEx dataset. RNA-sequencing raw counts are considered now. All $\sim$ 11000 samples available were considered at this time. Not surprisingly in the GTEx dataset it is retrieved the same

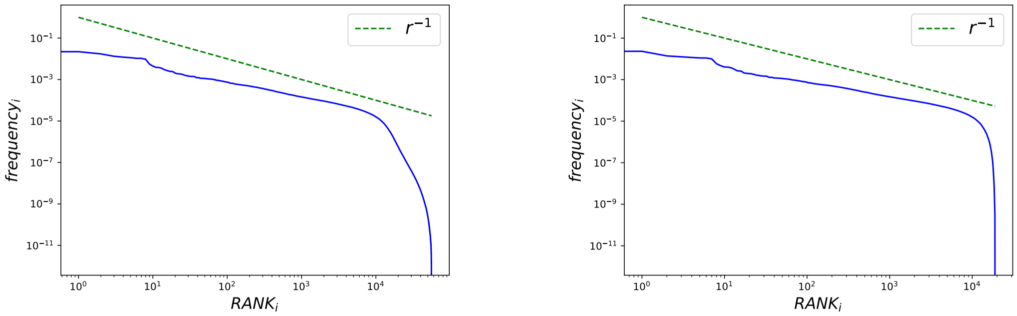


Figure 3.4: Zipf's law from GTEx count data. On the left, all genes are considered, on the right only protein-coding ones.

behaviour described before. The power law is found and considering non-coding genes leads to a knee in the power law.

Going further in the analysis it is possible to make a histogram of occurrences defined in Eq(3.1), also known as $U$s. Even in this kind of analysis, it is possible to

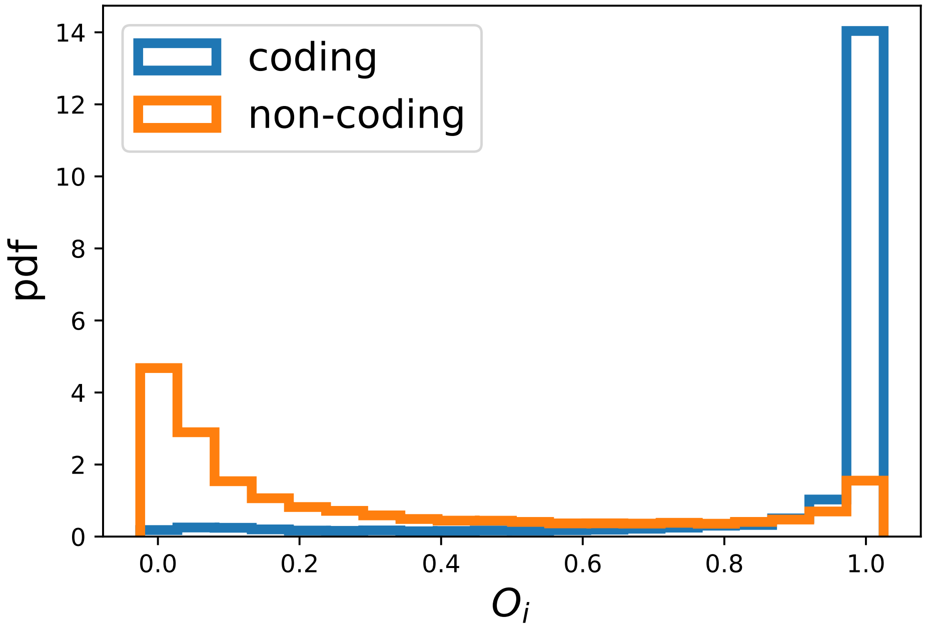


Figure 3.5: The histogram of the occurrences $O_i$. Coding and non-coding entries are coloured and normalized to 1.

see the different behaviour of coding and non-coding genes. The protein-coding genes express almost in every sample, so their occurrence is near to 1, non-coding genes are more specific, so they are present only in a subset of the dataset and many of them have a small occurrence.

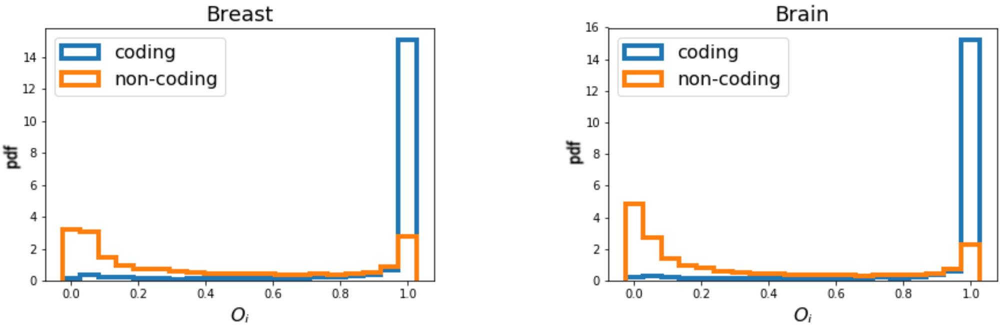


Figure 3.6: Differences in coding and non-coding occurrence distribution is observed also when looking at one tissue a time.

The same behaviour can be observed considering not all the samples but just the ones of a given tissue. In this case, $O_i = 0$ means that a gene has a non-zero expression in just one of the samples of the tissue considered; in other words, if a gene never expresses in a tissue it is not considered when constructing this tissue-specific $U$ distribution.

From now on, except were explicitly declared, analyses will be made considering protein-coding genes and counts with no normalization.

## 3.3 Null model construction

The data considered in this work comes from RNA-sequencing experiments. This kind of experiments uses wet biology methods to extract information from samples. If one imagines it exists an unknown function that describes the gene expression across the samples considered, what experimental people do is to sample this function, picking up some genes.

In this section it is described a null model of sampling. This is useful to verify if the data distributions seen are just an effect of this experimental sample or if they carry some useful and interesting information.

As described in [25], an ensemble of random matrix has to be created. This matrix is a collection of components and realizations exactly as the table in figure 3.1. The values of each component abundances in each realization $n_{ij}$ are randomly assigned with a probability determined by the global abundance in the whole dataset, Eq.(3.2). Values of each column are extracted until the size of Eq.(3.6) is reached. Strictly speaking it is a multinomial process with probability

$$P\left(\{n_i\}; M\right) = \frac{M!}{\prod_{i=1}^{N} n_i} \prod_{i=1}^{N} f_i^{n_i} \tag{3.8}$$

where $n_i$ is the number of times the component with frequency $f_i$ appears, being $f_i = \frac{a_i}{\sum_{c=1}^{N} a_c}$ as defined in Eq.(3.4).

Figure 3.7 shows an example of this, $M$ components are picked up concerning their frequency in the dataset. The most abundant components, which are also the ones with the highest frequencies (frequency is nothing but the normalized abundance), have a greater probability to be picked up. To gain statistics, an ensemble of these matrices is created and then averaged.

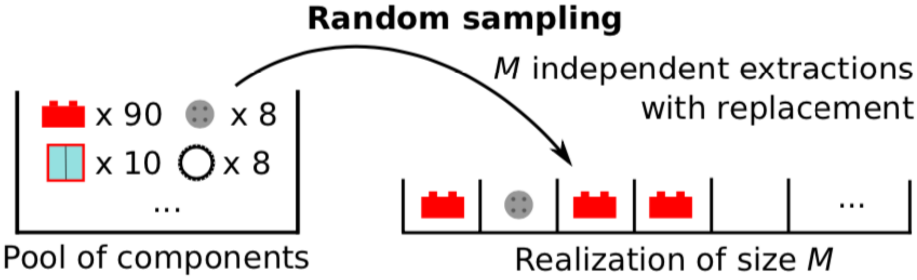


Figure 3.7: Random sampling of components to build a realization of size $M$.

Once null matrices are constructed, one can compare the real distributions with the ones obtained with this null model. The Zipf's law sampled is identical to the data's one, by definition. By construction, even the distribution of the sizes are identical in the real data and in the sampling matrices.

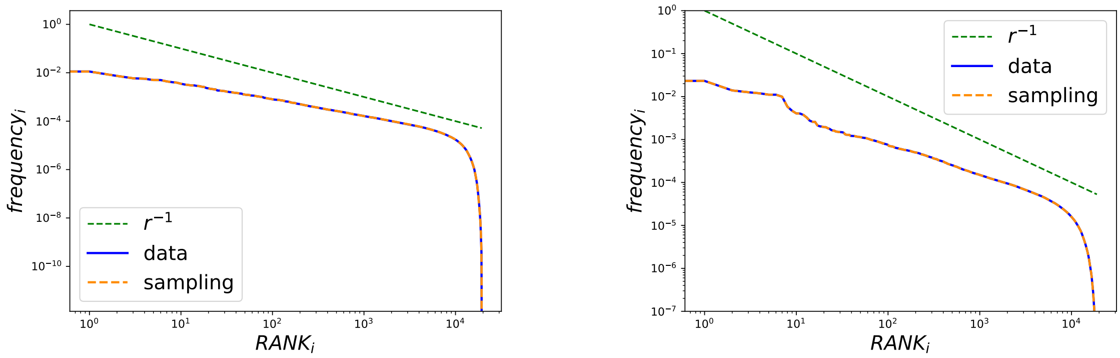


Figure 3.8: Zipf's law sampled. TCGA cancer data on the left and GTEx healthy samples on the right. By definition original frequencies and sampled ones are identical.

Looking at the $U$s in figure 3.10, it is evident that data behave differently from sampling. This is a signal that the null model is not enough to explain the structure of the data matrix. In particular, it is evident that the null model generates matrices with more components with high occurrence comparing to the original data. This can be easily explained: in the real world some genes are highly expressed but only in a subset of the whole dataset. These genes are specific for certain type of samples. The null model gets the information that such genes are highly expressed (they have a high abundance) and so picks these up quite often (components with high abundance have a greater chance to be picked up by the null model sampling). In figure 3.10 it is evident the difference between the real one and the sampled one: the null model data have a greater core.

Plotting on the abscissa the size of samples and on the ordinate the number of genes expressed, one point per sample, it is possible to obtain the so-called Heaps's law [26].

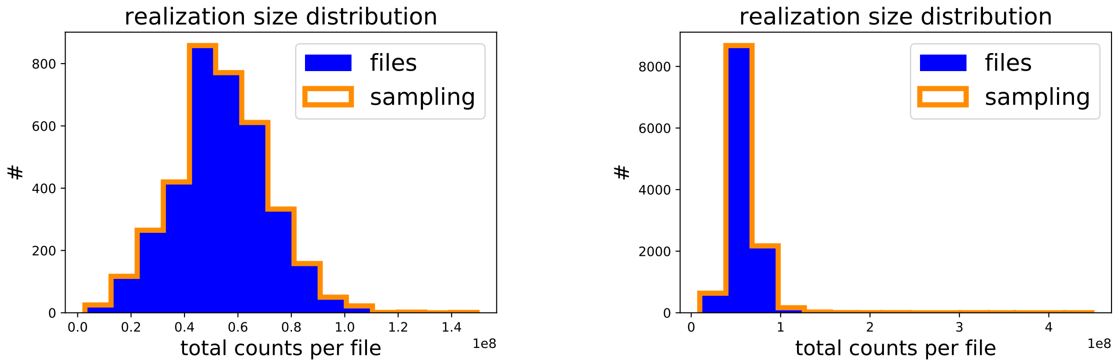


Figure 3.9: Distribution of size $M$. TCGA cancer data on the left and GTEx healthy samples on the right. By definition sampling and original sizes are identical.

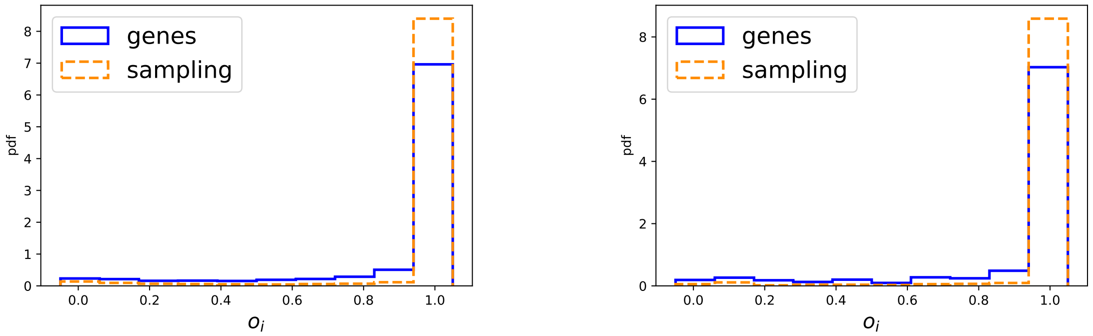


Figure 3.10: Occurrence distributions. TCGA cancer data on the left and GTEx healthy samples on the right. Sampling is reported for comparison.

In figure 3.11 the Heaps' law is presented compared to the one obtained by sampling. Again the curves differ and the null model is not enough to explain the trend. Note that each data point shares the abscissa with a sampling point (figures 3.9 are nothing but the histograms of the abscissas of figure 3.11). Moreover, it is interesting that the ordinate does not start from zero. This happens because there are a lot of genes that express everywhere, the core. It happens that the sampling curve is translated above the data's one. This means that to build a sample of size $M$ just by sampling it is necessary to use a greater number of different genes than the number of different genes actually expressed in nature. In other words in the real world only the genes that are really useful in a sample are expressed, and this is not describable just by a sampling model. This fact is coherent with the fact that the $U$s differ.

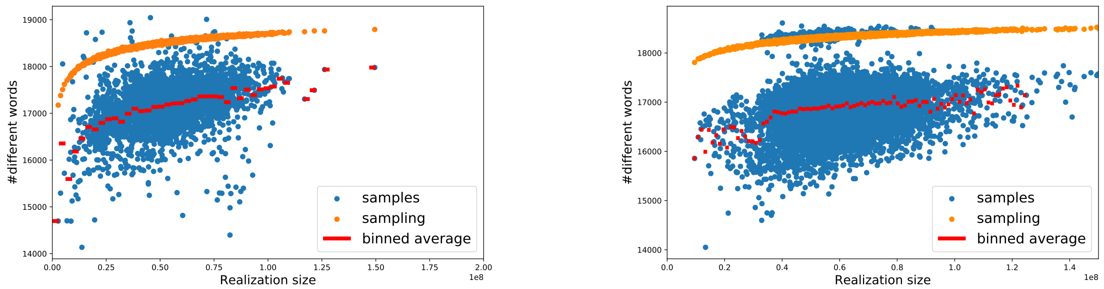


Figure 3.11: Heaps' law. TCGA cancer data on the left and GTEx healthy samples on the right. Sampling is reported for comparison.

Another way to see this is looking at the histograms of the number of different genes expressed, actually the distribution of the figure 3.11 y-axis. Figure 3.12 shows that these distributions are completely different, if one looks at the data and the sampling.

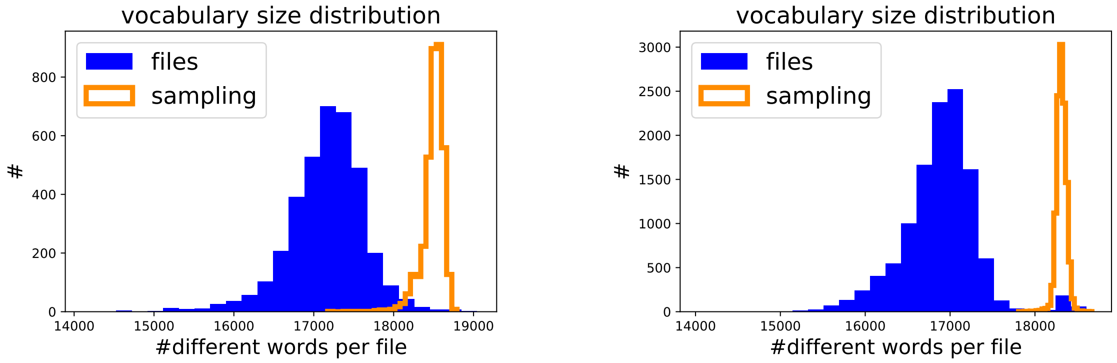


Figure 3.12: Number of genes expressed in a sample. TCGA cancer data on the left and GTEx healthy samples on the right. The difference between the original data and sampling is evident.

## 3.4 Statistical laws differentiate by tissue

Considering the GTEx dataset of healthy samples it is possible to study differences in tissues; [27] suggests some approaches.

First of all, it could be interesting to study which is the fraction of transcriptome that can be explained by a certain number of genes. To do this, firstly it is necessary to select all the samples of a given tissue. Then one estimates the average expression of each component (gene). At this point, one has the average abundance of each gene in a tissue, dividing by the sum of all the components it is possible to obtain the fraction of the total counts in the tissue due to each gene. Sorting from greater to smaller, integrating (cumulative summing) and normalizing, one has the fraction of transcript due to $1, 2, 3 \ldots$ genes. This is reported in figure 3.13.

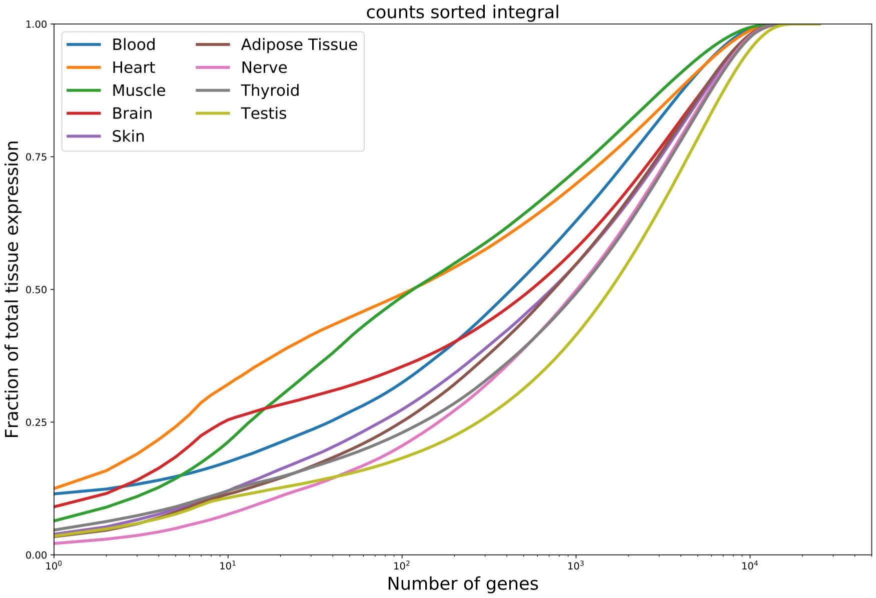


Figure 3.13: The integral of the sorted abundances for each tissue.

Here, when a curve is steep it means that a few genes' expression represents a great fraction of the total size of the transcriptome. If a curve is smooth it means that many genes are necessary to describe the whole transcriptome for that particular tissue. This analysis shows that different tissues have a different complexity in terms of the number of genes necessary to build the transcriptome (in average). In figure 3.14 the same analysis is done for the sub-tissues of brain, also these sub-types present a great separation.

Another way to interpret this analysis is thinking figure 3.13 as the integral of the Zipf's law. So it could be interesting to examine the Zipf's law one tissue a time. In figure 3.15 the Zipf's law for some tissues with an extremal behaviour are reported. From this point of view, each tissue has its particular slope. The steeper the Zipf the simplest the tissue: the transcript of a simple tissue can be described with a few genes.

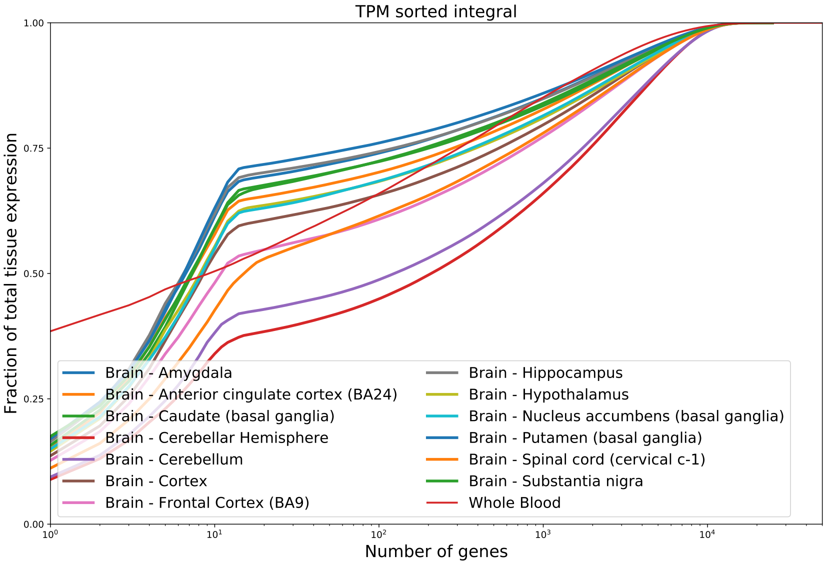


Figure 3.14: The integral of the sorted abundances for sub-types of brain. This is done using TPM to avoid biases due to gene lengths. Whole blood is plotted for reference.

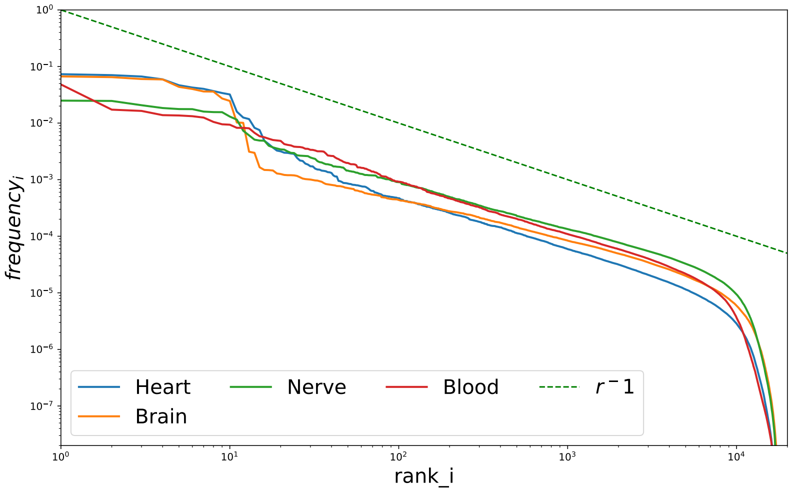


Figure 3.15: Different tissues present different power laws.

Coming back to the transcriptome analysis. In figure 3.13 the point where the curve reaches 1 corresponds to the total number of genes expressed, the remaining ones have a 0 expression and do not contribute to the transcript. This can be visualized again with the Heaps' law: the number of genes expressed seen in the Heaps' law plot is nothing but the number of genes necessary to explain the whole transcriptome. In figure 3.16, it is evident that there is some kind of tissue differentiation even when looking at the Heaps' law. In other words, two samples with the same size but of different tissues have a different number of genes expressed. The same analysis can be made by looking at the disease type of cancer samples. In this case, there is no evident differentiation as shown in figure 3.17. The only diseases that behave differently are *Parangliomas*, but these are associated only to brain, so the differentiation seen is just a brain separation. This means that separate diseases would be tricky and much more difficult than just separate tissues. The hierarchic approach described in the next chapters will be useful because it is able to separate tissues and disease types at different layers of the hierarchy.

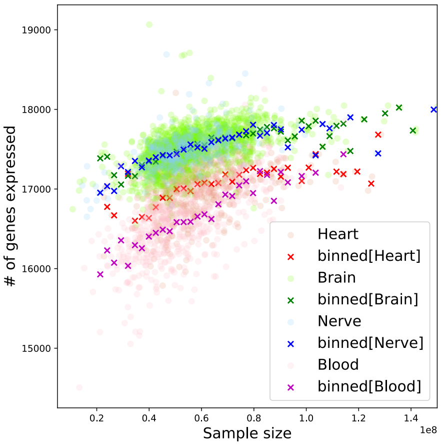


Figure 3.16: Different tissues express a different number of genes fixed the sample size. This plot was done using counts, using TPM it is not possible because the size on the x-axis is a would be a constant.

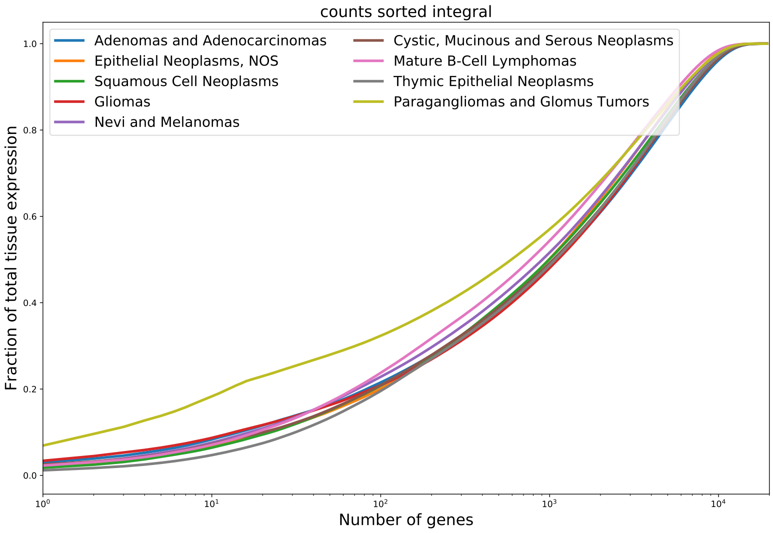


Figure 3.17: The integral of the sorted abundances for each disease type.

All these analyses suggest that there must be a sort of hidden structure in the data that is somehow related to the tissue each sample comes from. In particular, there are many Zipf's laws hidden behind the data and each sample is built looking at one of these. Moreover, given two samples with a similar size, it happens that the number of genes necessary to build that realization is not always the same (shown by Heaps' law) and it is somehow related to the sample's tissue.

**In conclusion** some interesting laws were found in the datasets considered in this work. Statistical laws from linguistics were identified and studied with different normalizations. Some interesting facts emerged: for example, the differences between coding and not-coding genes or the great core of protein-coding genes. The interesting fact, useful in the next sections of this work, is that behind the data there are different power laws and these are different for each tissue. In the next chapters, it will be discussed how to use this fact to train a model able to distinguish samples in a dataset.

# Chapter 4

# Scaling laws

One of the goals of this work is to search, reveal, study and use universal laws in gene expression data. Approaches from different field of science are considered at this point, as already done in chapter 3.

In can be interesting to study how the gene expression changes and what is the behaviour of genes across all the samples.

Given a matrix of components and realizations as the one in figure 3.1, with expression entries $n_{ij}$, one can select a row and estimate its mean $m_i = \langle n_{ij} \rangle_j$ and its variance $\sigma_i^2 = \langle n_{ij}^2 \rangle_j - \langle n_{ij} \rangle_j^2$. A row is nothing but a component that, in the data considered in this work, is represented by a gene. At this point, the analyses consider raw counts as entries.

### Variance versus mean

First of all, it could be interesting to study the variance of expression $\sigma^2_{\text{counts}}$ versus its average $\langle \text{counts} \rangle$ across tissues. In figure 4.1 the scatter plot of variance versus mean

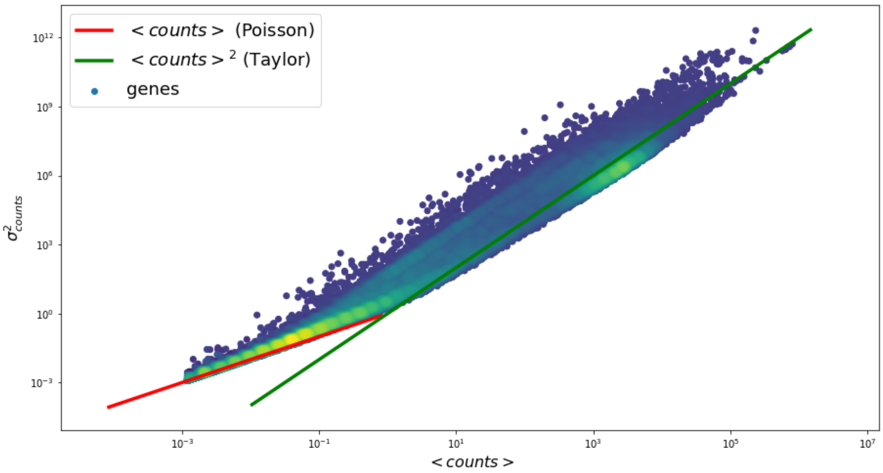


Figure 4.1: Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. All genes are considered.

reveals some interesting facts. First of all, it is evident that data have a double scaling behaviour: when the mean is small ($\lesssim 1$) data scale a Poisson-like ($\sigma^2_{\text{counts}} \sim \langle\text{counts}\rangle$), at higher means data present instead a quadratic scaling ($\sigma^2_{\text{counts}} \sim \langle\text{counts}\rangle^2$) known in ecology as Taylor's law [28]. This means that at low averages data's behaviour is due to the sampling process; on the contrary, Taylor's law reveals the non-trivial distribution across samples of the gene expression.

Another interesting fact is that looking at the density of points (colours in figure 4.1) two clouds of points emerge: one at low averages and one at high averages. These correspond to coding and non-coding genes, remembering section 3.2 these two kind of genes have different behaviours: protein-coding genes are highly expressed in the majority of the samples, non-coding ones are less expressed (and so less sampled) in a few samples.

### Coefficient of Variation

A similar analysis, common in literature, is the analysis of the coefficient of variation squared $CV^2 = \frac{\sigma^2_{\text{counts}}}{\langle\text{counts}\rangle^2}$ represented in figure 4.2. The behaviour is complementary to the one discussed above; a double scaling, quite common in the literature looking at single-cell RNA sequencing data [29], is present. Even looking at $CV^2$ it is evident the presence of the protein-coding and non-coding clouds of points. The non-coding genes have a Poisson-like scaling, $\sigma^2_{\text{counts}} \sim \langle\text{counts}\rangle$ so $CV^2 = \frac{\sigma^2_{\text{counts}}}{\langle\text{counts}\rangle^2} \sim \frac{1}{\langle\text{counts}\rangle}$, otherwise the protein-coding genes are on the Taylor-like curve $CV^2 = \frac{\sigma^2_{\text{counts}}}{\langle\text{counts}\rangle^2} \sim$ constant.

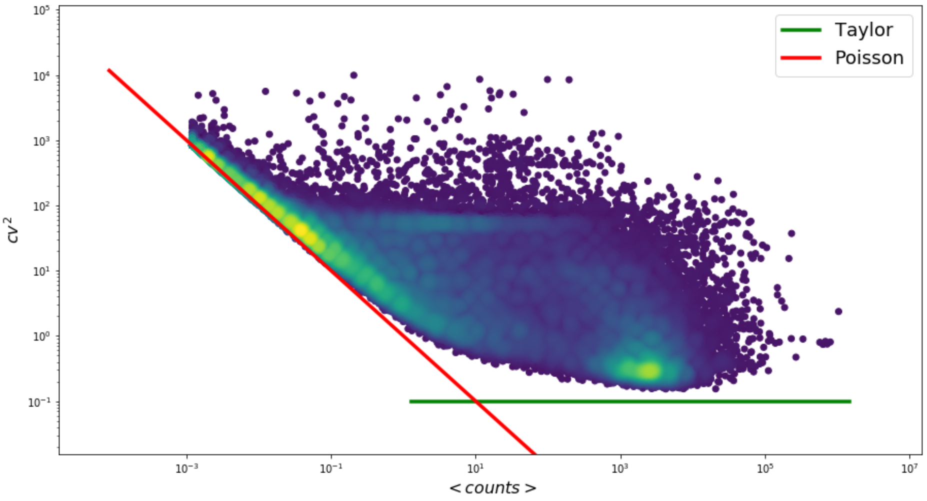


Figure 4.2: Coefficient of variation squared versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. All genes are considered.

**Protein-coding genes** can be isolated and considered on their own. The same analysis confirms that the cloud of points on the Taylor-like scaling is made by protein-coding genes.

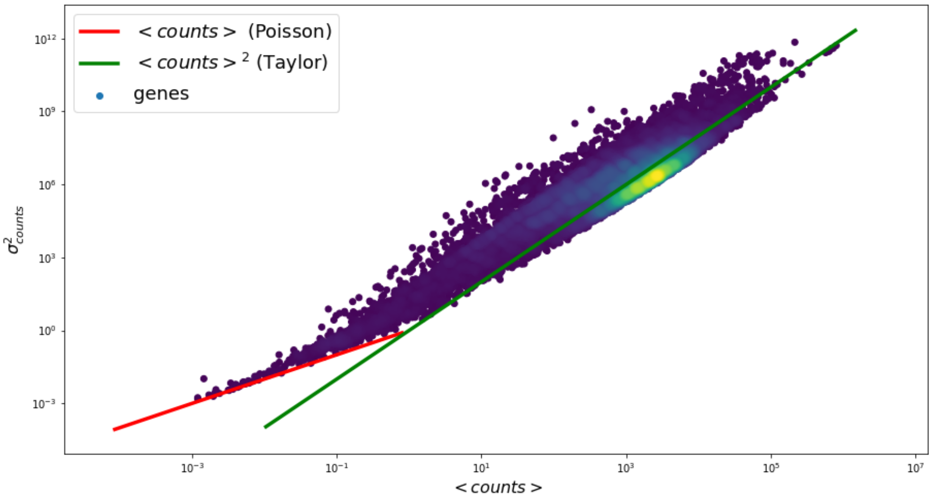


Figure 4.3: Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. Only protein-coding genes are considered.

Following the sampling model of [25] summed up in section 3.3 the averages and variances can be estimated on null matrices. In figure 4.4 the comparison between real genes and sampling data. The sampling has got a double scaling as well; this is quite interesting, it means that the global scaling is due to the Zipf distribution and the sizes' distribution themselves, they are, by definition, identical in the data and in sampling. Moreover, the sampling points draw a lower bound of the data, this encodes the information that the data are more variable (have higher variance) than just sampling, so there must be some biological information hidden that causes this over-variable behaviour.

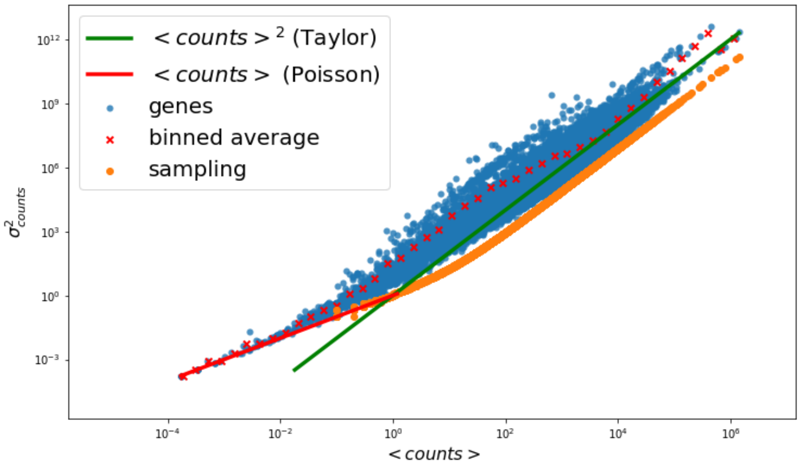


Figure 4.4: Variance versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. In orange the sampling components. Only protein-coding genes are considered.

Again it is possible to analyse the $CV^2$, this time considering only protein-coding genes. Figure 4.2 confirms that the cloud of points near the Taylor-like scaling is made of protein-coding genes and a double scaling is seen once again.

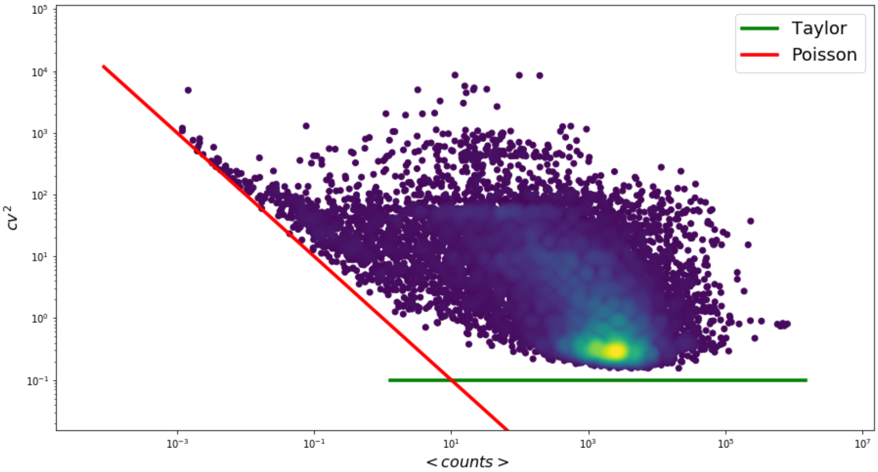


Figure 4.5: Coefficient of variation squared versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. Only protein coding genes are considered.

In figure 4.6 the same plot compared to the sampling data. The double scaling is evident also for the sampling points. Note that $CV^2$ has got a lower bound at 0 which corresponds to the less variable case: all expressions are identical in all samples ($\sigma^2_{\text{counts}} = 0$). There is an upper bound at $R-1$, with $R$ the number of realizations, that corresponds to the most variable case: a component expresses in only one realization and is 0 elsewhere.

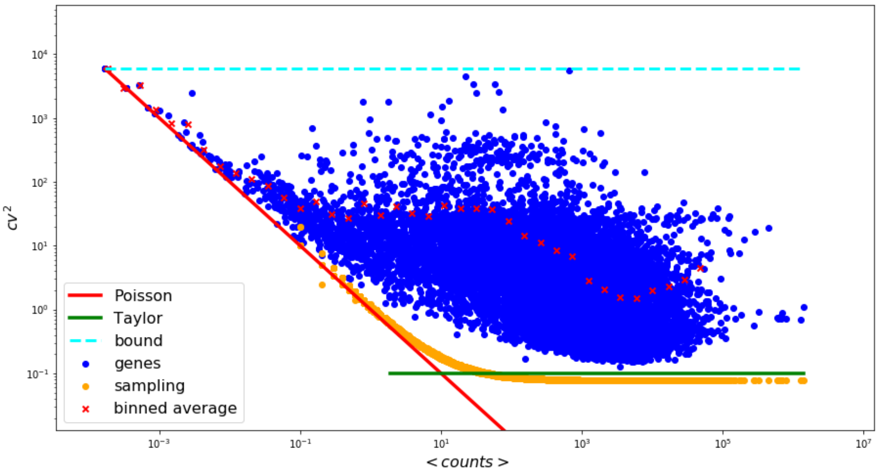


Figure 4.6: Coefficient of variation squared versus average. In red the Poisson-like scaling, in green the Taylor-like scaling. In orange the sampling components.

Finally, the data have a double scaling when looking at their global variance across realizations, a Poisson-like scaling in the region where the sampling experimental process is more important and a Taylor-like scaling where the complexity of the data emerges. Non-coding genes have got low expression and are rare; protein-coding genes, otherwise, express a lot and everywhere and carry more information; this behaviour results in a double scaling. All genes are more variable than a sampling null model and this is the evidence that something interesting is hidden behind the data.

**Average versus occurrence**
Another interesting analysis can be the study of the relationship between the occurrence and the expression average. In figure 4.7 the result is shown: it is clear that there is a relationship between occurrence and average, genes that express in more realizations (higher occurrence and right in the figure) have a higher average. Moreover, there aren't genes that have high expression in few realizations; rare genes are also difficult to find so have a small average. Note that the average has got a bound because counts are integer numbers, so if, for example, one gene express in $n$ of the $R$ samples, it has occurrence $O_i = \frac{n}{R}$ and its average can't be lower than $\langle \text{counts} \rangle = \frac{1*n}{R}$

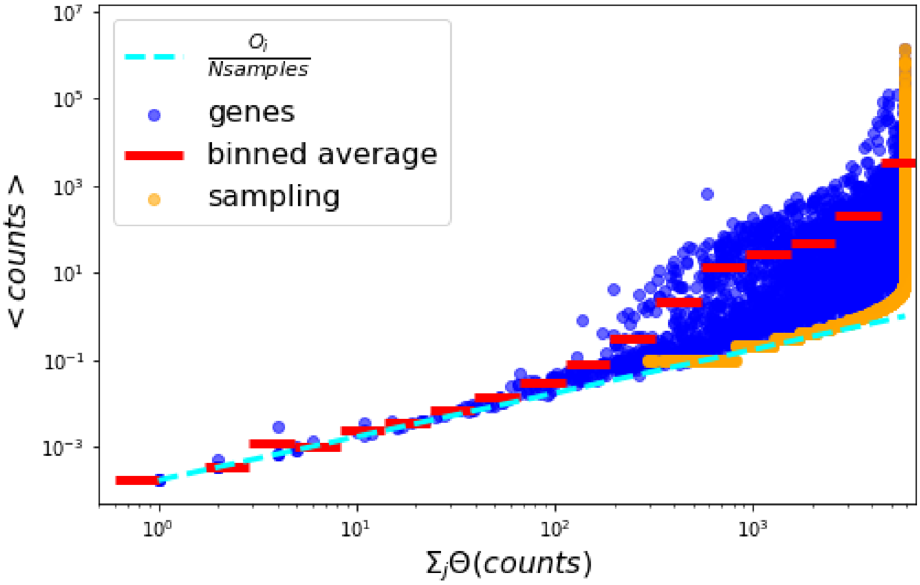


Figure 4.7: Relationship between the occurrence of a gene and its average across realizations.

**In conclusion** the study of gene expression across samples reveals interesting facts. Coding and non-coding genes have different behaviours. The null model isn't, again, enough to explain the data. Real data genes revealed themselves as more variable than expected.

The analyses described in this chapter will be useful in the next part of the work: in fact, they empower the possibility to define a gene selection model to isolate highly variable genes if necessary.

# Chapter 5

# Topic modelling

Once extensively analysed the structure of the dataset, the goal becomes to develop a machine learning method which learns the hidden structure of the data.

Remembering chapter 3, there it emerged some kind of structure behind data: each tissue seemed to be sampled by a different power law. A topic modelling approach is here proposed. Topic modelling has been developed and studied to approach linguistics problems, so this algorithm was developed considering a network of words and books in input, links represent the abundance of a word in a book. In chapter 3, it was evident that there are many similarities between data considered in this work and linguistics' corpora. Referring to data used in this project **samples** will be the documents and **genes** will be the words. It is expected that topics represent some properties of the system due to the gene expression distribution in samples.

The idea is that behind the data there are hidden variables that describe the relationship between the genes and the samples. Let's call these variables topics. Firstly it is necessary to build a bipartite network of genes and samples, then nodes are linked considering the gene expression value in the dataset.

The output of this kind of models consists of sets of genes, the topics, with a probability distribution $P(\text{gene}|\text{topic})$ and probability distributions of these topics inside each sample $P(\text{topic}|\text{sample})$, together they give the relationship between a *sample* and a *gene*.

In this work, an innovative and recent approach to topic model is proposed. The algorithm was presented by [5] and [30] explained it in details. This model is an evolution of a stochastic block model [31]. It is called hierarchical Stochastic Block Model (hSBM).

The ultimate goal is being able to separate healthy and diseased samples, then find and separate well-known tumour types and finally extend the actual knowledge and retrieve the tumour sub-types.

One of the advantages of this particular algorithm is that it is hierarchical, so it applies community detection at different layers of resolution. So the output has got different resolutions and a different number of clusters at each layer. One extreme layer is, by definition, the one which separates genes ($\simeq$ words) and samples ($\simeq$ samples) in two blocks. Then the number of clusters increase going deeper in the hierarchy. The other extreme layer is the one where the number of clusters is comparable with the

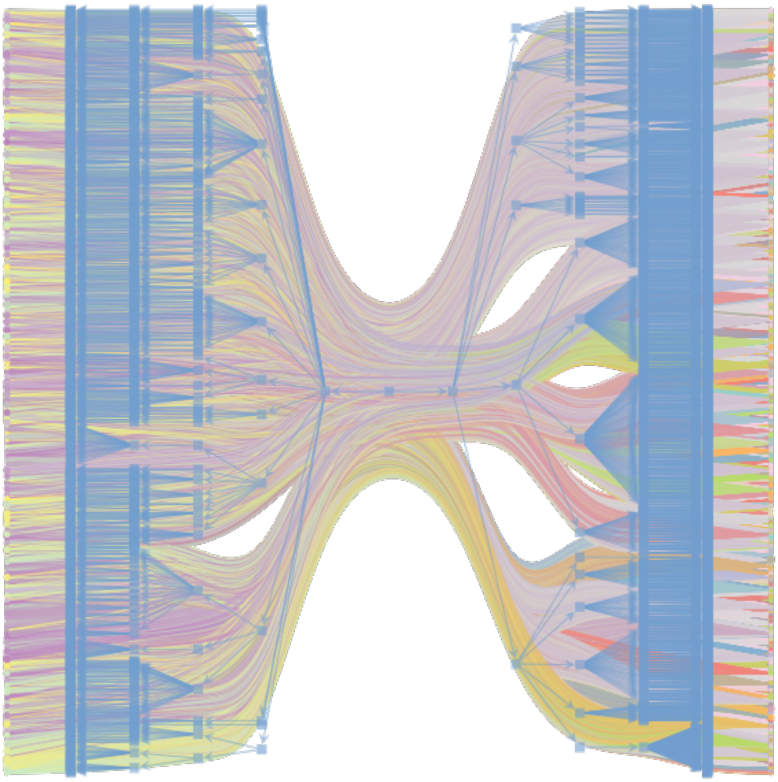


Figure 5.1: An example of a bipartite network. Samples are on the left, genes are on the right. Each link is weighted by the gene expression value. On the left side, all nodes of the same colour are clusters of samples. On the right side, all nodes with the same colour are a set of genes, also known as topics.
Blue lines represent the cluster structure, each blue squared-dot is a set of nodes, lines delineate the hierarchical structure.
It is clear, in the middle, the network separation between genes and samples.

number of nodes.

What the algorithm does is to run a sort of Monte Carlo simulation and find the best partition of the data. The probability that the hidden variables $\theta$ describe the data $G$ $P(\theta|G)$ can be written as a likelihood times a prior probability as

$$P(\theta|G) = \frac{P(G|\theta)\overbrace{P(\theta)}^{prior}}{\underbrace{P(G)}_{\sum_\theta P(G|\theta)P(\theta)}}.$$

It is possible to define a description length

$$\Sigma = -lnP(G|\theta) - lnP(\theta),$$

so that $P(\theta|G) \propto e^{-\Sigma}$. Description length is the quantity of information needed to describe the model. Following the Occam's razor the shorter the description length

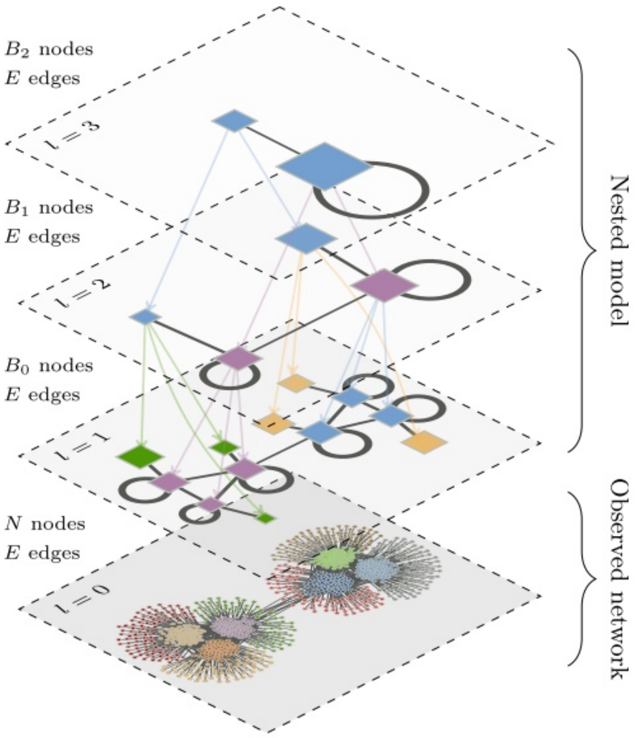


Figure 5.2: Example of a hierarchical structure. At $l = 0$ the number of cluster is comparable with the number of nodes, is the situation with many small clusters. Then they're merged in bigger clusters at other layers of the hierarchy.

the better the model. Moreover, the likelihood $P(G|\theta)$, can be written as $\frac{1}{\Omega}$ where $\Omega(\theta)$ is the number of networks that is possible to build given $\theta$. This corresponds to a microcanonical ensemble with entropy $S = Ln\,(\Omega)$. According to [32] entropy $S$ can be written as

$$S = \frac{1}{2}\Sigma_{r,s} n_r n_s H\left(\frac{e_{rs}}{n_r n_s}\right),$$

where $n_r$ is the number of nodes in the block $r$, $e_{rs}$ the number of links between nodes of group $r$ and nodes of group $s$ and $H$ is the Shannon entropy $H(x) = xLog_2(x) + (1 - x)Log_2(1-x)$. Note that $S$ is minimal if $\frac{e_{rs}}{n_r n_s}$ is close to zero, $r$ and $s$ are two completely separated blocks or if it is close to 1, $r$ and $s$ are groups with many connections; this allows finding groups with nodes very disconnected or topic and clusters with a lot of connections. Note that the description length of a network's state is related to the entropy of the states' ensemble

$$\Sigma = S - lnP(\theta).$$

The algorithm tries to minimize $S$, so that $\Sigma$ is minimized, so $e^{-\Sigma}$ is maximized, but this is $P(\theta|G)$ that is the required probability to maximize.

The Monte Carlo simulation works in a few steps:

- a node $i$ is chosen;

- the group of $i$ is called $r$;
- a node $j$ is chosen from $i$'s neighbours, the group of $j$ is called $t$;
- a random group $s$ is selected;
- move of node $i$ to group $s$ is accepted with probability $P(r \to s|t) = \frac{e_{ts}+\epsilon}{e_t+\epsilon B}$;
- if the move to $s$ is not accepted, a random edge $e$ is chosen from group $t$ and node $i$ is assigned to the endpoint of $e$ which is not in $t$;

in figure 5.3 an example of these steps.

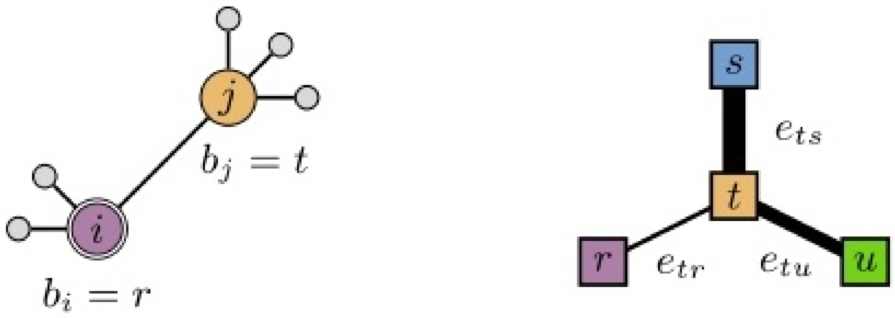


Figure 5.3: Left: Local neighbourhood of node $i$ belonging to block $r$, and a randomly chosen neighbour $j$ belonging to block $t$.
Right: Block multi-graph, indicating the number of edges between blocks, represented as the edge thickness.
In this example, the attempted move $bi \to s$ is made with a larger probability than either $bi \to u$ or $bi \to r$ (no movement), since $e_{ts} > e_{tu}$ and $e_{ts} > e_{tr}$.

In order to remove eventual biases due to the initial configuration the model is run with 5 different initial states, then the final state with the minimal entropy is selected.

Once the model runned, it is possible to estimate the probability distribution of words inside a topic

$$P(w|t_w) = \frac{\#\text{ of edges on } w \text{ to } t_w}{\#\text{ of edges on } t_w}$$

and the topic distribution inside a document

$$P(t_w|d) = \frac{\#\text{ of edges on } d \text{ from } t_w}{\#\text{ of edges on } d}.$$

This algorithm can be set to accept overlapping partitions; in this case, the presence of a word in a topic is non-trivial and can be estimated as

$$P(t_w|w) = \frac{\#\text{ of edges on } w \text{ to } t_w}{\#\text{ of edges on } w}.$$

The membership of a document in a cluster is

$$P(t_d|d) = \frac{\#\text{ of edges on } d \text{ to } t_d}{\#\text{ of edges on } d}.$$

See appendix A for a detailed analysis of the maths behind the algorithm and https://hub.docker.com/r/fvalle01/hsbm for the extension of [5] to non-linguistics component systems datasets.

## 5.1 Test the model with metrics and benchmarks

Before running topic modelling, it is useful to define some metrics to test and benchmark the model. In particular the model searches sets on the two sides of the network: the one containing samples and the one containing genes. Samples are extracted from datasets where much metadata are available, some of these metadata labels will be used to benchmark the model. To study genes, enrichment test are instead necessary.

Looking at the samples side of the network, the outputs are sets of samples, let's call these clusters. One can state that the model works if all, or at least the majority, of samples in the same cluster share some label. Here the sample primary site is considered as the main label.

Note that this is a non-supervised model, nevertheless a ground truth is available from metadata. So every sample has a certain probability to have a certain property (the true tissue label), let's call this $P(C)$ and a certain probability of being in a cluster (model's output), let's call this $P(K)$. It is possible to define, for instance, the homogeneity

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{5.1}$$

defining the entropy

$$H(C|K) = \sum_{c \in \text{labels}, k \in \text{clusters}} \frac{n_{ck}}{N} Log\left(\frac{n_{ck}}{n_k}\right) \tag{5.2}$$

where $n_{ck}$ is the number of nodes of type $c$ in cluster $k$, $N$ the number of nodes and $n_k$ the number of nodes in cluster $k$. It is evident that if all nodes inside cluster $k$ are of the same type $c$ $n_{ck} = n_k$, $H(C|K) = 0$ and $h = 1$, it is actually a full homogeneous situation.

Another quantity can be defined: the so-called completeness

$$c = 1 - \frac{H(K|C)}{H(K)}, \tag{5.3}$$

$H(K|C)$ is defined in the same way as Eq.(5.2). Completeness measures how well nodes of the same type are distributed in the same cluster.

Ideally one wants a method which output is both homogeneous and complete. So it is possible to define the V-measure as the harmonic average of the two:

$$\text{V} - \text{measure} = 2\frac{hc}{h+c}, \tag{5.4}$$

which is actually the normalized mutual information between $P(C)$ and $P(K)$ [33]. Please refer to appendix B for the detailed maths. In table 5.1 a simplified example of the homogeneity and completenesses ideas. [34] proposed a similar metric to compare topic modelling algorithms' performances.

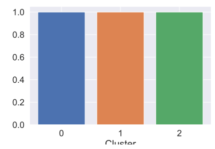
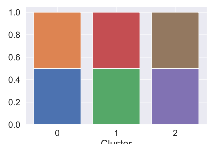
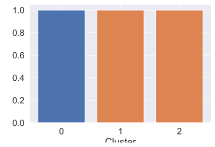
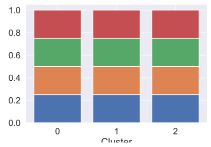
| | Homogeneous | Not homogeneous |
|---|---|---|
| Complete |  |  |
| Not complete |  |  |

Table 5.1: Examples of homogeneity and completeness. Homogeneous clusters contain all nodes with the same label. A label is complete if it is fully represented by a single cluster. In this image some extreme examples of these definitions.

In figure 5.4 an example of the V-measure score estimated at the different layers of the hierarchy; note that the number of clusters increases going deeper in the hierarchy. In the same figure homogeneity and completeness are reported, note that with few clusters the situation is more complete, but when the number of clusters increases completeness goes down and homogeneity increases. This happens because if a cluster is small it is easier to fulfil it with similar objects. On the other side if one has few clusters it is easier to complete them putting similar objects in the same cluster.

In order to validate the model, its scores will be compared to the ones obtained with standard approaches such as standard hierarchical clustering and a classical topic model approach using Latent Dirichlet Allocation.

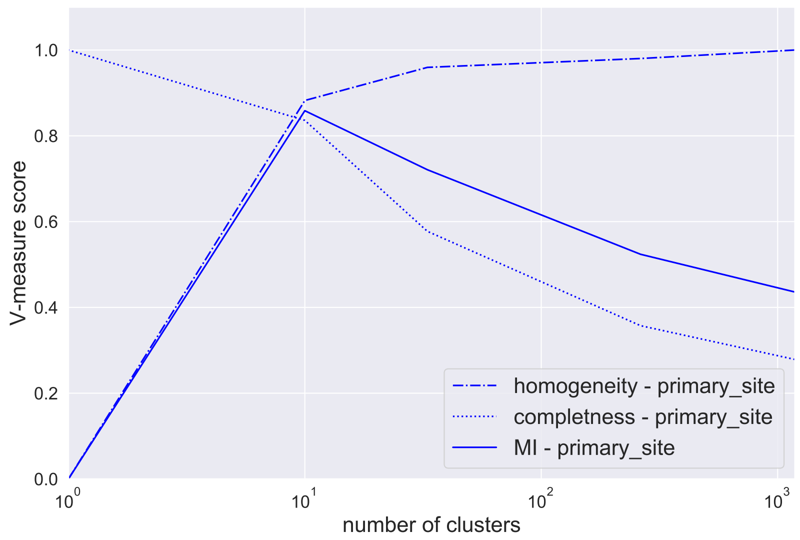


Figure 5.4: Score across hierarchy. The V-measure or normalized mutual information MI is the harmonic average between homogeneity and completeness.

### 5.1.1 Hierarchical clustering

The first algorithm considered for comparison is hierarchical clustering. Hierarchical clustering is a general family of clustering algorithms; they build nested clusters by merging or splitting them successively. This hierarchy of clusters can be represented as a tree or dendrogram. The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample. To perform hierarchical clustering and merge the samples successively the `AgglomerativeClustering` function from *scikit-learn* was used. Note that this approach was applied just to the samples, nothing was done to classify genes. In this case, genes are only dimensions in which samples are represented.

The `AgglomerativeClustering` object performs a hierarchical clustering using a bottom-up approach: each observation starts in its cluster, and clusters are successively merged. A linkage criterium determines the metric used for the merging strategy.

First of all, distances between all elements are estimated; in this work standard euclidean distance was used. Then elements are merged using the linkage criterium; in this work standard *Ward linkage* was used. The Ward linkage minimizes the sum of squared differences between the distances. It is a variance-minimizing approach. Here the specific configuration used in this work:

```
from sklearn.cluster import AgglomerativeClustering
AgglomerativeClustering(
    affinity='euclidean',
    compute_full_tree='auto',
    linkage='ward',
    n_clusters=x,
    )
```

note that the number of clusters is a free variable `x`: hierarchical clustering is not able to determine the ideal number of clusters. This was set from the output of the hierarchical Stochastic Block Model. In figure 5.5 an example of hierarchical clustering. Note that close nodes (with the euclidean distance defining *close*) are linked firstly.

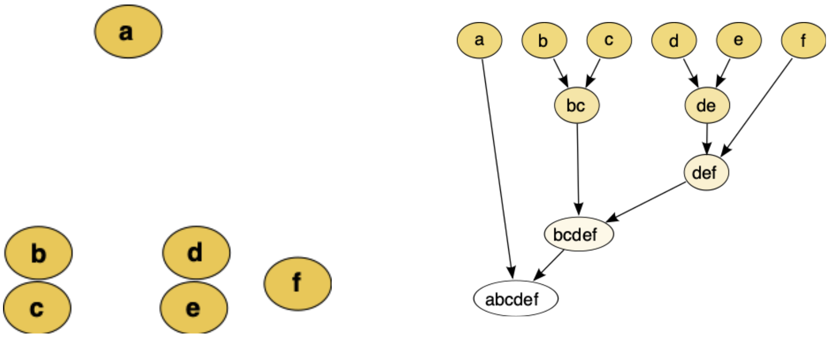


Figure 5.5: Example of hierarchical clustering. Nodes are merged creating a tree (on the right).

### 5.1.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation is the standard and well-known approach to topic models. It has got more restrictive priors than hierarchical Stochastic Block Model and needs some parameters to be set. It uses some different methods to maximize the posterior probability to observe some latent variables given the data. As well described in [2] LDA is a generative model and can be summarized as follows:

- set the number of topics $K$ and the parameters $\eta$ and $\alpha$
- for each topic $k$ generate $\beta_k \sim \text{Dirichlet}(\bullet|\eta)$
- for each document $d$ generate $\theta_d \sim \text{Dirichlet}(\bullet|\alpha)$
- for each word in $d$
  - generate $z \sim \text{Multinomial}(\bullet|\theta_d)$
  - generate $w \sim \text{Multinomial}(\bullet|\beta_k)$

this process is represented in figure 5.6. The goal is to maximize the posterior probability

$$P(w, z, \beta, \theta|\alpha, \eta) = \prod_{d=1}^{N} P(\theta_d|\alpha) \prod_{n=1}^{N_d} P(w_{dn}|z_{dn}, \beta_k) P(z_{dn}|\theta_d) \prod_{k=1}^{K} P(\beta_k|\eta) \tag{5.5}$$

where

- $N$ is the number of documents;
- $K$ is the number of topics as set by the user;

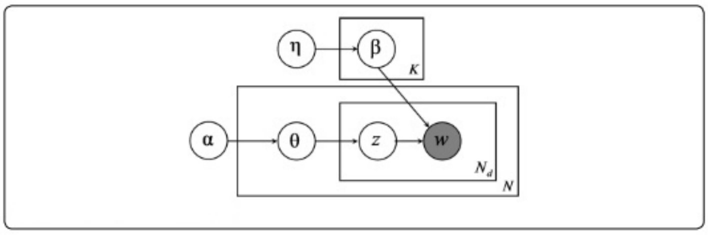


Figure 5.6: LDA schematic representation.

- $w$ are words;
- $N_d$ is the number of words in document d;
- $\alpha$ and $\eta$ are parameters of the model (usually $\eta = 0.01$ and $\alpha = 50/K$);
- $P(\theta|\alpha)$ and $P(\beta|\eta)$ are Dirichlet distributions.

When the distributions $\beta$, $\theta$ and $z$ are estimated, the outputs are the topic distribution in documents $P(\theta|\alpha)$ and the word distribution in topics $P(\beta|\eta)$. Again the number of topics $K$ is not found by the model and needs to be set, this has been got from hierarchical Stochastic Block Model's output.

## 5.2 Preprocessing and filtering the dataset

This work aims to use the hierarchical Stochastic Block Model (hSBM) described before. Data considered in this work have $\sim$ 11000 samples as documents and $\sim$ 60000 genes as words ($\sim$ 20000 if one considers only protein-coding genes). The original paper [5] considers 63 Wikipedia articles along with 3140 words. The great amount of data requires some tricks to filter the network and make the computation faster. Different approaches were tested to pre-process the data. All of them involve the quantities defined in chapter 3. The goal is to identify components which can best partition the realizations and, in the meanwhile, isolate the most interesting genes.

**Low occurrence genes** were selected firstly to approach topic modelling. A 0.5 threshold was set on occurrence and only genes with $O_i < 0.5$ were considered. This method selects genes that appear (have expression greater than zero) only in less than half samples. This approach has got some limitations, for instance, genes that appear everywhere (with occurrence $\simeq 1$) but change their behaviour across realizations are not considered.

**Tf-idf (term frequency–inverse document frequency)** should help. This approach doesn't take into account original expression values $n_{ij}$, but a transformed version

$$n_{ij}^{new} = \frac{n_{ij}}{M_j} \times (1 - Log\,(O_i))$$

which increases the importance of components with small occurrence $O_i$. This is widely used in linguistics to wash out common words. Tf-idf doesn't select or reduce the number of components (genes), which is still an issue. Moreover, the number of link between nodes is no longer an integer so an approximation is necessary and this introduces a bias.

**Highly variable** genes can be selected. This is done using the $CV^2$ analysis done in chapter 4. Plotting the coefficient of variation versus the mean, one point for each component, one can reveal components that have a variance higher than components which, on average, have similar behaviour. Binned averages and variances were estimated. Only genes with a $CV^2$ over a $\sigma$ were considered. This method seems useful to select genes even if the binned average bound is quite noisy.

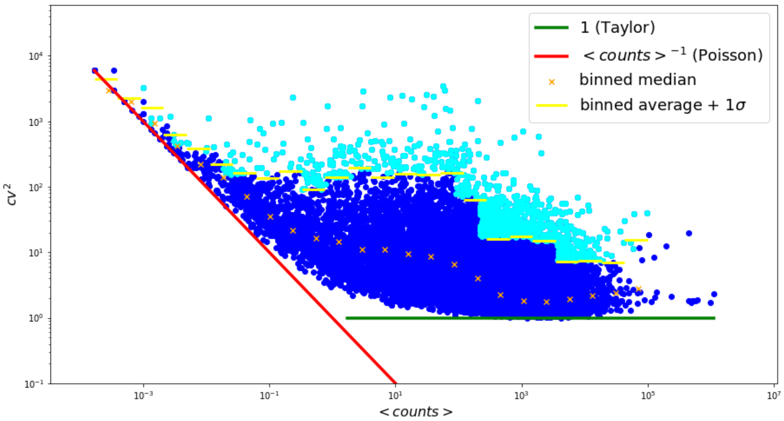


Figure 5.7: Highly variable genes. In cyan genes that are 1$\sigma$ over the average of their bin.

**Distance from boundaries** can be a similar and alternative method to select highly variable genes. In this case, the boundary is both smooth and well-defined. The distribution as discussed in 4 has got both a Poisson-like and a Taylor-like boundary. So it is possible to consider only the components that are the most distant from these boundaries. Moreover, these boundaries can be found with a simple null model. As shown in figure 4.6, the sampling model defines the lower bound of the data.

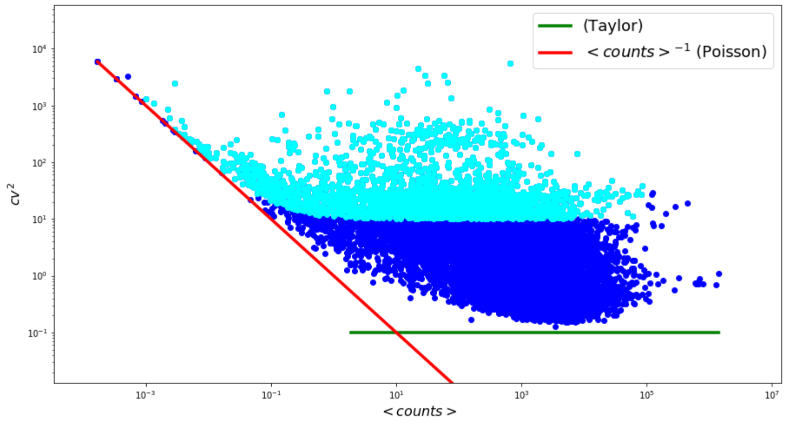


Figure 5.8: In cyan genes which distance from boundaries is greater than $10CV^2$.

The last two approaches are the ones which lead to the better results, in the following section gene selection was done by getting only highly variables genes. To reduce the number of documents, samples are picked up uniformly random from a subset of all the available ones.

## 5.3 Run

### 5.3.1 Run on Gene Tissues Expression dataset

Once the model was set and adapted to RNA-Sequencing data, it was run on a subset of the GTEx dataset. A subset of samples was chosen randomly to reduce the computing time needed. The analysis hereby described took about 2 days to be run on a 16 core CPU, 100GB memory facility. The great amount of memory is needed to temporary store the network configuration at each step of the Monte Carlo simulation.

First of all, to rapidly have information about the interest on the oncoming result the metric above described were considered. In figure 5.9 it is represented the V-measure score versus the number of clusters found at different layers. The result is quite good, the maximum score is over 0.8. Considering that, for example, [1] obtained a similar score analysing similar dataset considering just homogeneity, this can be considered a quite good result. A second interesting fact is that both the tissue label (primary site) and the sub-tissue label (secondary site) obtain such a good score. Moreover, the secondary site score's peak is at a higher number of clusters coherently with the fact that there is a greater number of sub-tissue labels. This score can be useful to extract the correct level of the hierarchy the consequent analysis should be made on.

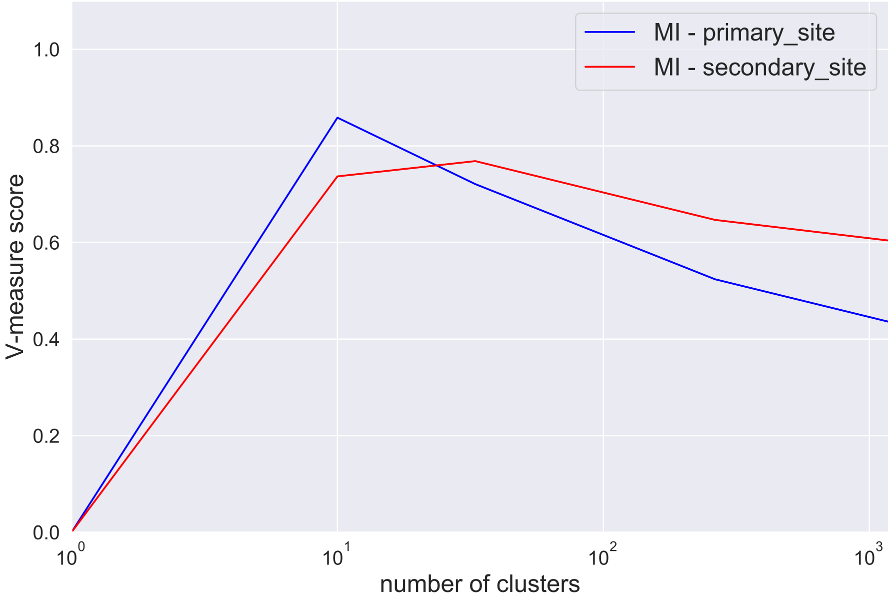


Figure 5.9: Scores across the hierarchy. The performances classifying the primary site and secondary site are compared. Note that with one cluster the completeness is 1 but the homogeneity is 0 so the score goes down.

In figure 5.10 the relationship between the clusters at different layers is evident. Each row is a layer of the hierarchy and arrows represent the path of a node across the hierarchy. Note that clusters don't overlap and the separation done at the greater level is maintained across all the hierarchy. This representation gives an idea of the point of the hierarchy where the tissues separate, see in figure 5.10 cluster 2 that splits in two clusters separating *breast* and *adipose tissue*.

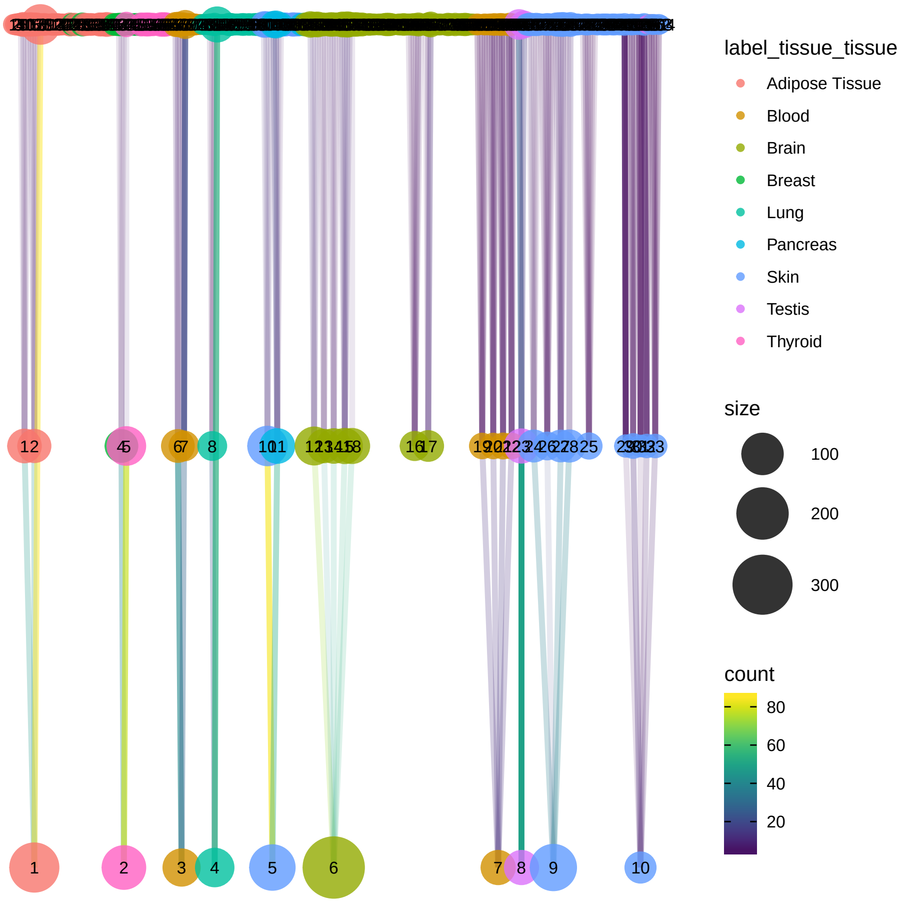


Figure 5.10: Hierarchy of the files' nodes. In the top the layers where the output consists of many small clusters, in the bottom the output with few big clusters. The colour of the nodes refers to the most present tissue in that cluster. Arrows represent how nodes pass from a hierarchy layer to another. The bigger the balls, the bigger the cluster. The more yellow the link the more nodes are in common between the clusters of the two different layers. Plotted using clustree [35].

In figure 5.11 each column is a cluster and each colour is a tissue of the dataset. It is evident that the majority of the tissues are identified: the first, second, fourth, fifth, sixth, seventh, and tenth columns are fully and uniformly coloured of the same colour. These correspond to an identification of *brain*, *skin*, *lung*, *blood* and *testis*. In the normalized representation of the same clusters the homogeneity of the clusters is more evident. Going deeper in the hierarchy and looking at a layer with more cluster the result, shown in figure 5.13, demonstrates that, at this point, all the tissues are

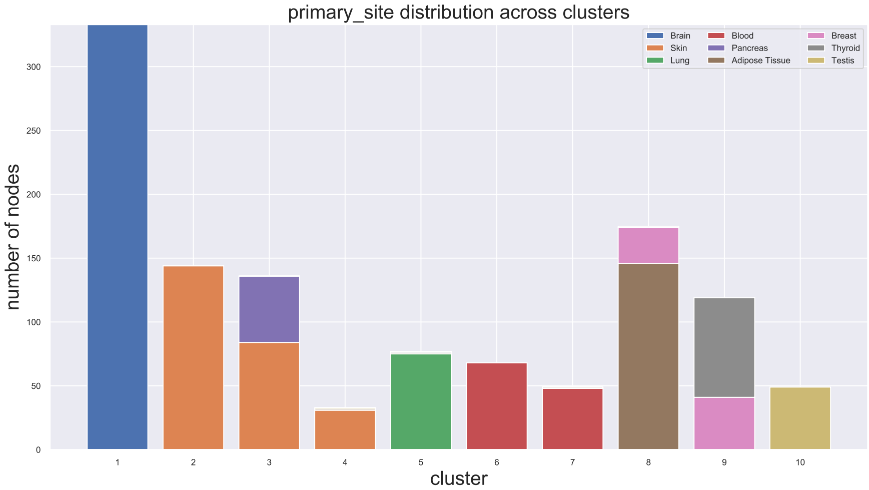


Figure 5.11: Clusters composition at the level of the hierarchy with the higher score. Each column is a cluster, each colour is a label.

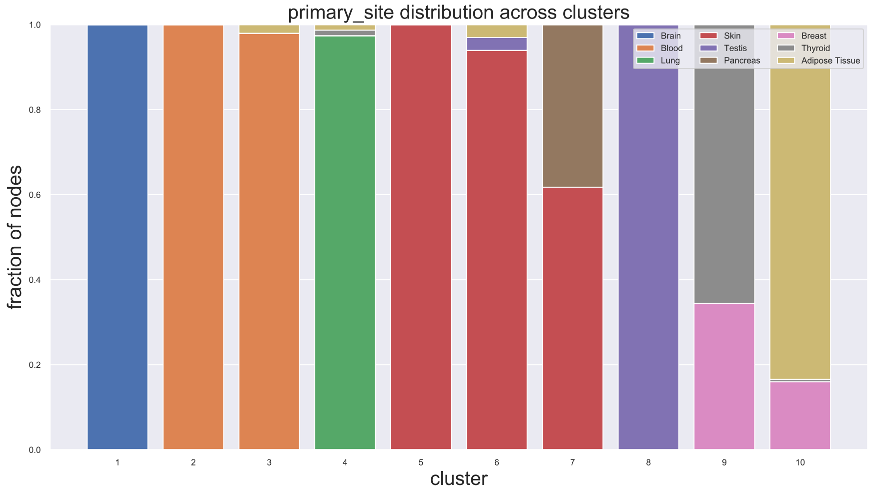


Figure 5.12: Normalized composition of clusters. Again each column is a cluster, each colour is a label.

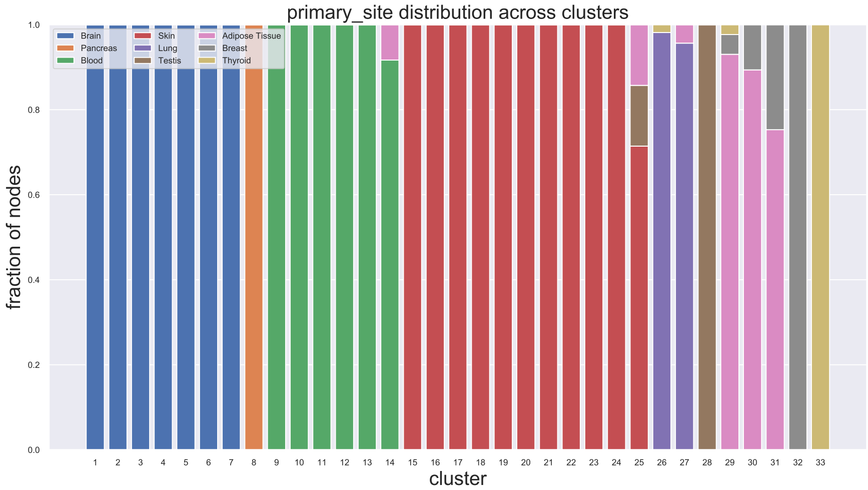


Figure 5.13: Normalized composition of clusters at a deeper level.

separated and each cluster is full of nodes sharing the same tissue. Even looking at sub-tissues the results is quite good. It is not always easy to separate all the sub-parts of the *brain*, nevertheless, the *cerebellum* is well identified (column 27) and *blood* is distinguished in *whole blood* (columns 4-8) and *lymphocytes* (column 3).

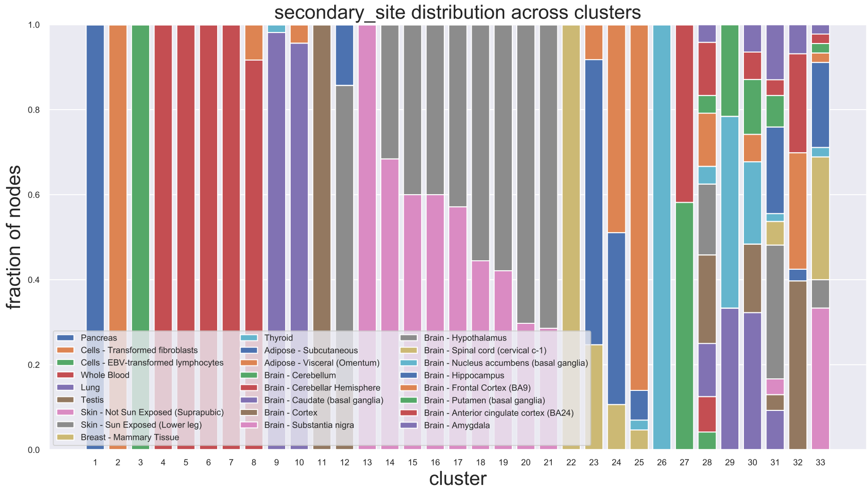


Figure 5.14: Normalized composition of clusters for the secondary site sub-tissue labels.

### 5.3.2 Null model shuffling labels

A null model of cluster composition is necessary if one would be able to state that a result is better than expected. This was done by doing the same analysis but reshuffling the labels of the nodes. Reshuffling was done exchanging the label of each node with the one of another node picked up uniformly random. Doing so the number of clusters and the cluster sizes are maintained. In figure 5.15 an example of clustering with random labels, it is evident that all clusters have similar and homogeneous composition. Note that not every tissue has the same number of samples, so, for example, *brain* is more represented than other tissues.

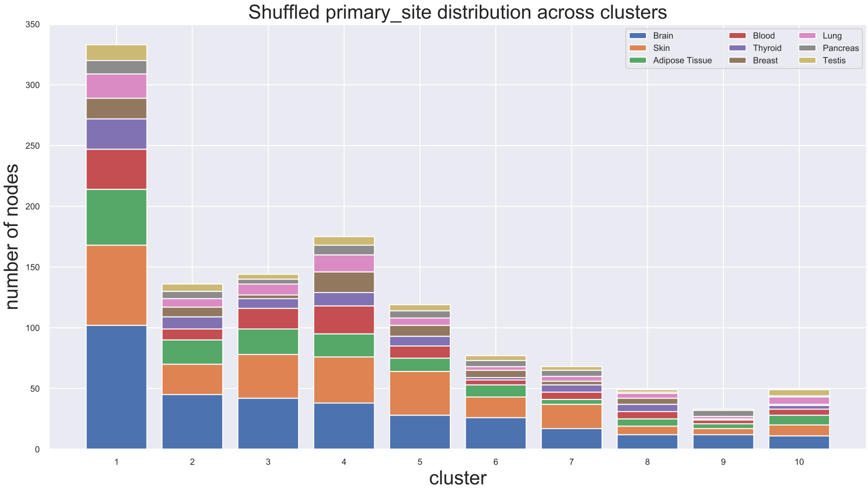


Figure 5.15: Example of visualization of clusters with reshuffled labels.

All the results described in the previous pictures are quite qualitative. To have a more objective and mathematical measure of the success of the algorithm it is possible to measure the fraction of the most representative label in each cluster $k$

$$max_{l \in labels} \left( \frac{n_{lk}}{n_k} \right)$$

being $n_{lk}$ the number of nodes labelled $l$ in cluster $k$ and $n_k$ is the number of nodes in cluster $k$. This is represented in figure 5.16 for the level where the V-measure is maximized (the best results are expected at this resolution). In figure 5.16 on the left is shown the most representative label fraction for each cluster, on the right the histograms of the same quantity. Models' clusters are very homogeneous with the majority of cluster full, almost 100%, of the same tissue. It is also clear that reshuffling the labels the result is very different and so it is possible to admit that the models behave better than expected.

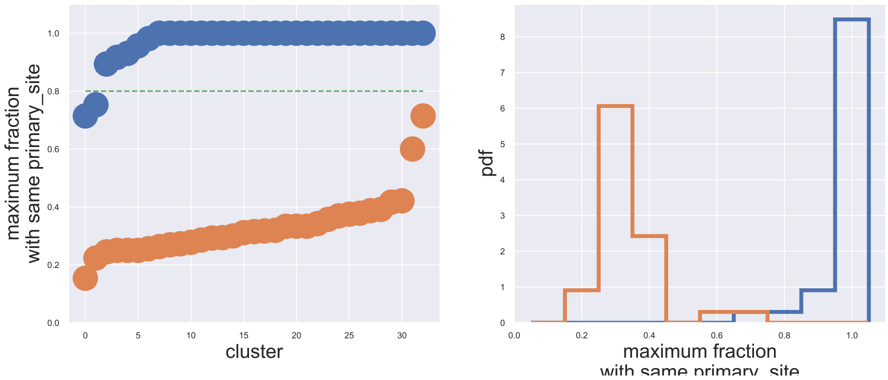


Figure 5.16: The fraction of cluster composed by the representative label versus cluster. On the right the distributions of this measure.

In figure 5.17 the same analysis is done for every level of the hierarchy. It is interesting to notice that at the deepest level (upper left in the figure) the random reshuffling and the real labels have the same behaviour. This happens because, at this level, clusters are very small and so it is easier to pick up nodes with the same label (in the extreme case of a cluster with size 1 it is always full of the same label). This shows that the deepest level is not interesting: results are the same with random labels; moreover the reshuffling null model is good to show up eventual biases due to small cluster sizes.

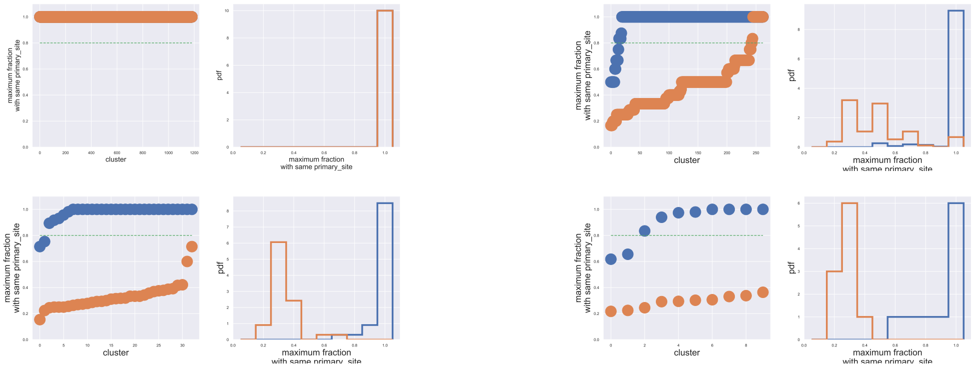


Figure 5.17: The fraction of the most representative label in all clusters for different levels of the hierarchy. From upper left the deepest layer than downright the superficial one.

A similar analysis can be made considering not just the number of the cluster but the cluster size, this is shown in figure 5.18. It is interesting to notice that the shuffle null model and the real labels clusters are different, so there must be some kind of signal. Clearly the model is able to output even big clusters full of the same label (points upper right in figure 5.18). In figure 5.19 the same analysis for all the levels of the hierarchy. It is interesting to see how going up in the hierarchy the two signals become different, as shown before the deepest layer (upper left in the image) is not

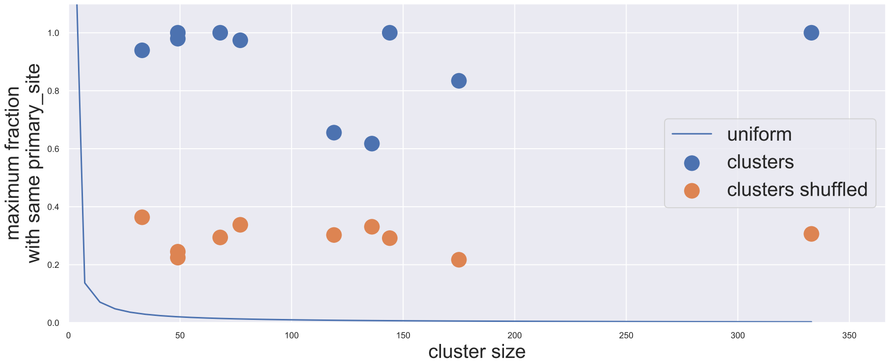


Figure 5.18: The fraction of the most representative label versus cluster size.

different from null model and so it is not interesting.

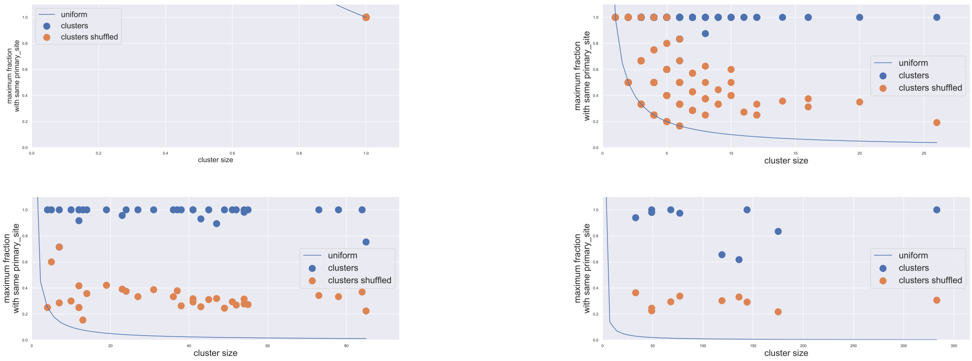


Figure 5.19: The fraction of the most representative label versus cluster size across the hierarchy. From upper left the deepest layer than downright the superficial one.

At this point to deeper investigate the structure of the clusters, it can be interesting to study how many labels are present in each cluster. The fraction of the most represented label defined above carries no information of what happens to the remaining labels. For example, if one cluster is composed of 80% by label **A** and 20% by label **B** and another cluster is composed 80% by label **A**, 10% by label **B** and 10% by label **C** they have both a fraction of maximum representative label 80% but the second in this example is more heterogeneous. Counting the number of different labels in each cluster can reveal this kind of effects. In figure 5.20 it is represented the number of different labels versus cluster size. It is evident that the reshuffling case is quite different from the real one, almost every cluster in the null model has got every label. It is interesting to notice that the model outputs even big cluster with one label. In figure 5.21 the same analysis for all the layers of the hierarchy. Even here the deepest level does not differ from the null model. Nevertheless, in layers with higher V-measure score, there is a strong signal that the reshuffling model is quite different from the model's output.

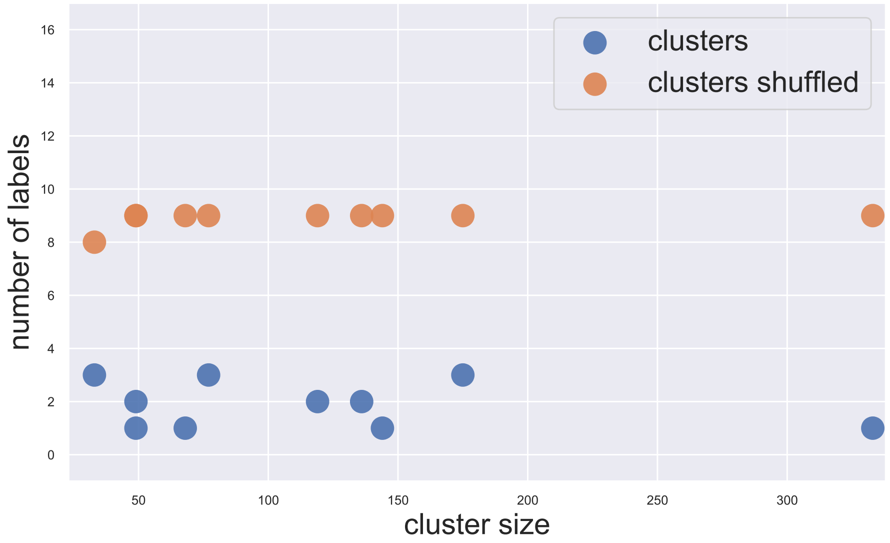


Figure 5.20: The number of different labels in each cluster versus cluster size.

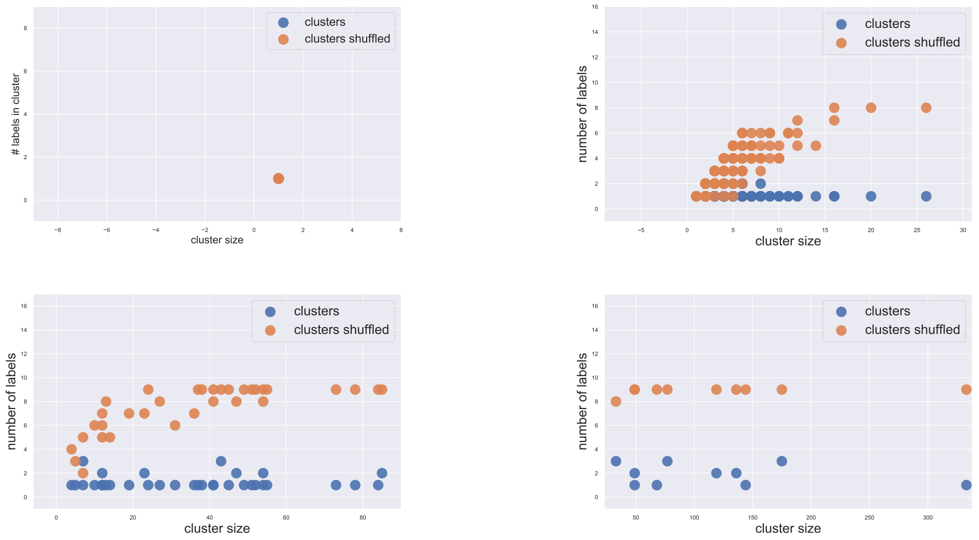


Figure 5.21: The number of different labels in each cluster versus cluster size. From upper left the deeper layer than downright the superficial one.

Having constructed the null model it is possible to estimate the V-measure score also for the null model. The results are reported in figure 5.22. Moreover, remembering the V-measure or normalized mutual information defined in Eq.(5.4) it is possible to estimate a mixed score which considers the homogeneity of the primary site and the completeness of the secondary site, doing so the score goes up if going deeper in the hierarchy the model makes more cluster with the same tissue but separates sub-tissues. It is not a big deal if one loses completeness regarding tissues (the model separates one big cluster full of the same label into two small ones) but gain information at the next resolution. This becomes clear if one looks at the big *blood* cluster that in the next level of the hierarchy is separated into two clusters of *blood*, one of *whole blood* and one of *lymphocytes*. The result is that this mixed score is the highest one.

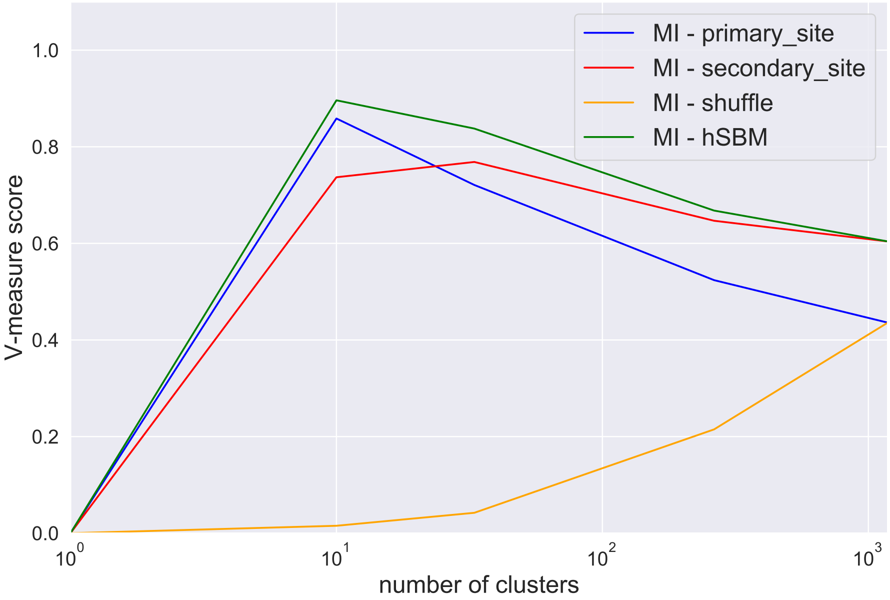


Figure 5.22: Scores across the hierarchy. The scored is compared with some random labels. In blue the score for the primary site labels, in red for the secondary site labels, in yellow the shuffled labels, in green the mixed score with primary homogeneity and secondary completeness.

### 5.3.3 Comparison with standard algorithms

At this point, it was verified that the model has got interesting output: it reaches high scores and has got a strong signal against the null model, at least at some levels of the hierarchy. It is now interesting to compare it with standard and well-studied similar algorithms. First of all, a comparison is made with hierarchical clustering. This is done using the standard scipy package [8], the metrics used was the euclidean one and the linkage method was set to Ward as briefly introduced in section 5.1.1. This is quite fast, it needs a couple of minutes on a dual-core, 8GB memory machine. In figure 5.23 the comparison between these scores, the hierarchical algorithm performs worse than hierarchical Stochastic Block Model and as highly expected better than the random model.

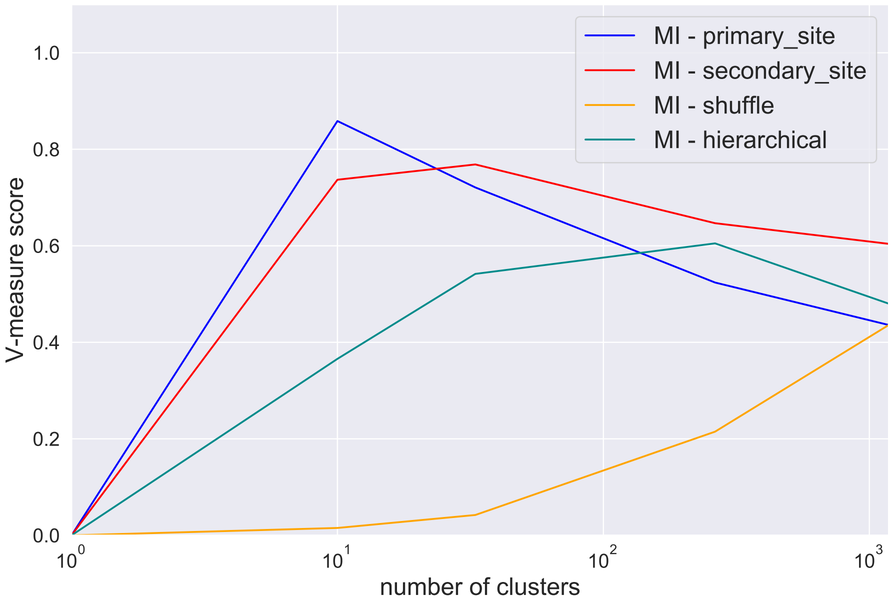


Figure 5.23: Scores across the hierarchy. The score obtained with hSBM is compared with hierarchical clustering and shuffled labels.

Another very used and well-studied algorithm is Latent Dirichlet Allocation briefly described in 5.1.2. Running LDA, as implemented standard scipy package, is quite fast and is comparable with hierarchical clustering in terms of CPU time. Note that once LDA package extracts the topics, it is necessary to define some clusters, to do so a standard Agglomerative clustering approach was used, the distance was set to *euclidean* and the linkage to *Ward*. In figure 5.24 the V-measure scores for all the algorithms described until this point are reported. It is clear that the hierarchical Stochastic Block Model performs better than all the others, LDA obtains a little worse score and hierarchical clustering is the worst of the three. It is highly expected that all models are quite different from the random model. The fact that hSBM and LDA have higher scores suggests that a topic model approach can be very useful in this kind of problems. Note that LDA and hierarchical clustering models were not fine-tuned and default parameters were used. Maybe a fine-tuning of these packages can lead to better and more satisfying results. This analysis, considering that the comparison was made with hierarchical Stochastic Block Model which is non-parametric and needs no setting, was done without any fine-tuning and standard parameters were set. This fact reveals another good point of hSBM, it extracts not only better clusters, but also the parameters necessary to this kind of models. Moreover, the number of topics was set to the one obtained from hSBM; LDA is not able to select the number of topics.

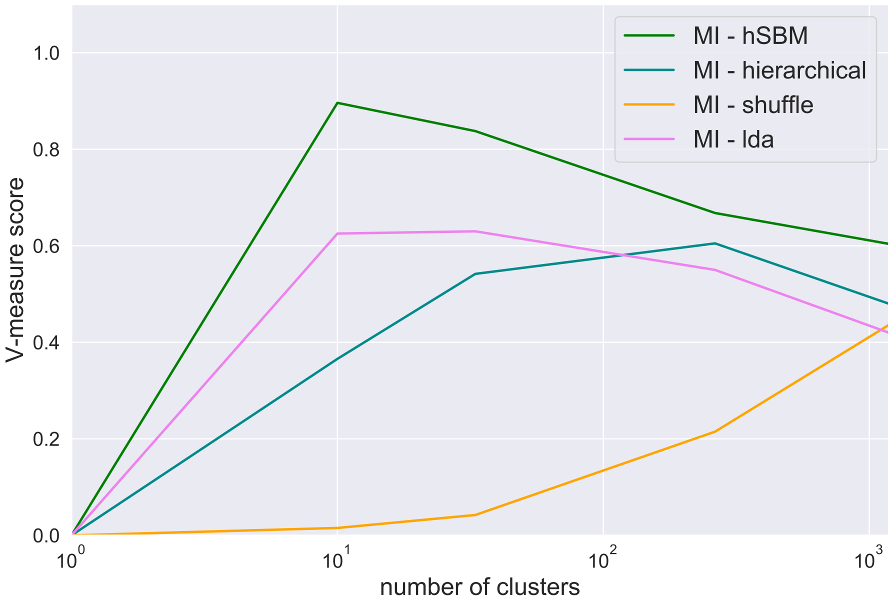


Figure 5.24: Scores across hierarchy for all algorithms used in this work.

### 5.3.4 Topics enrichment tests

The analyses up to this point considered only one of the two sides of the bipartite network; nothing was told yet about the genes (words in topic models). The model outputs some clusters of genes, though. From now on these clusters of genes will be called topics.

If one has got a set of genes, it is possible to perform an enrichment test to catch any important information and discover if there are any biological meanings behind it. Enrichment analysis checks whether an input set of genes significantly overlaps with annotated gene sets. In this work tests were made using Gene Set Enrichment Analysis (GSEA) [36] python tool [37], which performs a Fisher exact test (hypergeometric test). The Benjamini-Hochberg adjusted P-values is reported. Genes' annotation terms were searched in the following sets:

- GO[1] Molecular Function 2018;
- GO Biological Process 2018;
- GO Cellular Component 2018;
- Human Phenotype Ontology;
- Tissue Protein Expression from Human Proteome Map;
- KEGG 2019 Human;

[1] Gene Ontology

- NCI-60 Cancer Cell Lines;
- GTEx Tissue Sample Gene Expression Profiles up;
- GTEx Tissue Sample Gene Expression Profiles down;

in particular the two latter contain annotation specific for GTEx dataset [38].

In tables 5.2, 5.3 and 5.4 examples of enrichment test results. Each table is a topic by hSBM. On the results it is put a P-value cut at 0.05 and terms are sorted by the adjusted P-value. Tests were made on the topics at the level of the hierarchy which obtained the higher V-measure score on the sample side of the network. These results are very interesting, these enrichment tests demonstrate that not only the sample side of the network is well clustered but also the topics have a non-trivial meaning.

| Term | Adjusted P-value | Gene set |
|---|---|---|
| pancreas male 60-69 years | 1E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas female 40-49 years | 3E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 40-49 years | 5E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 30-39 years | 1E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas female 20-29 years | 1E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 50-59 years | 1E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas female 30-39 years | 1E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 50-59 years | 2E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 40-49 years | 2E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 30-39 years | 2E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 50-59 years | 2E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas female 20-29 years | 2E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 40-49 years | 3E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas female 50-59 years | 4E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 50-59 years | 4E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 50-59 years | 4E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas female 60-69 years | 5E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas female 50-59 years | 5E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 50-59 years | 5E-18 | GTEx Tissue Sample Gene Expression Profiles up |
| pancreas male 30-39 years | 6E-18 | GTEx Tissue Sample Gene Expression Profiles up |

Table 5.2: Enrichment test of a topic. It is clear the enrichment for pancreas-related gene sets.

| Term | Adjusted P-value | Gene set |
|---|---|---|
| brain female 40-49 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain male 50-59 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 60-69 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 60-69 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 60-69 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 40-49 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 40-49 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 60-69 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain male 60-69 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain male 50-59 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain male 50-59 years | 6E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain male 60-69 years | 7E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain male 50-59 years | 7E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain male 20-29 years | 7E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 60-69 years | 8E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 60-69 years | 8E-05 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 60-69 years | 1E-04 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 60-69 years | 1E-04 | GTEx Tissue Sample Gene Expression Profiles up |
| brain female 60-69 years | 1E-04 | GTEx Tissue Sample Gene Expression Profiles up |
| brain male 60-69 years | 1E-04 | GTEx Tissue Sample Gene Expression Profiles up |

Table 5.3: Enrichment test of a topic. It is clear the enrichment for brain-related gene sets.

So also the topics are related to the tissues and somehow are tissue-specific. In the next examples, the relationship between the topics and the samples will be further

| Term | Adjusted P-value | Gene set |
|---|---|---|
| blood male 50-59 years | 3E-23 | GTEx Tissue Sample Gene Expression Profiles up |
| blood male 50-59 years | 3E-23 | GTEx Tissue Sample Gene Expression Profiles up |
| blood male 40-49 years | 3E-21 | GTEx Tissue Sample Gene Expression Profiles up |
| blood male 60-69 years | 9E-21 | GTEx Tissue Sample Gene Expression Profiles up |
| blood male 40-49 years | 3E-20 | GTEx Tissue Sample Gene Expression Profiles up |
| blood female 60-69 years | 4E-20 | GTEx Tissue Sample Gene Expression Profiles up |
| blood male 60-69 years | 4E-20 | GTEx Tissue Sample Gene Expression Profiles up |
| blood female 50-59 years | 5E-20 | GTEx Tissue Sample Gene Expression Profiles up |
| blood female 50-59 years | 1E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood male 60-69 years | 1E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood male 60-69 years | 1E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood female 60-69 years | 1E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood male 60-69 years | 2E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood male 50-59 years | 2E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood female 40-49 years | 2E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood female 40-49 years | 2E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood female 60-69 years | 2E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood male 30-39 years | 3E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood female 50-59 years | 5E-19 | GTEx Tissue Sample Gene Expression Profiles up |
| blood female 60-69 years | 5E-19 | GTEx Tissue Sample Gene Expression Profiles up |

Table 5.4: Enrichment test of a topic. It is clear the enrichment for blood-related gene sets.

investigated. In particular following what was done by [18] the importance of each topic inside each sample, the $P(\text{topic}|\text{sample})$, will be discussed.

Separate healthy tissues is a good exercise and a good benchmark for models, but the real goals would be being able to classify diseased samples. It is not always easy to identify and classify cancer tissues. In particular, being able to separate tumour sub-types would be the ideal pursuance of this work. So let's switch to the analysis of cancer samples.

### 5.3.5 Run on The Cancer Genome Atlas

The same pipeline described so far can be applied to other datasets. In this section, the hSBM model is run on some samples from the TCGA. The principle is the same, but here samples come from cancer tissues, so there must be more complexity and variability behind the data. Moreover, being able to separate cancer samples is not always easy clinically and develop a method to do this can be fascinating and useful for the scientific community [1].

First of all, let's take a look at the V-measure scores. As shown in figure 5.25 the maximum score is $\simeq 0.8$, which is quite good, comparable with the healthy GTEx scenario. The publishers of the dataset [1] obtained similar score considering just homogeneity. In this dataset, there isn't a sub-tissue label as before, but a *disease type* cancer information is available. The disease type separation happens but obtain a lower score; the fact that there is no evident difference between Zipf's laws when separating data by disease type (previously shown in figure 3.17) means that all genes contribute to define this specific label. The hierarchic approach which separates firstly tissues and then cancer type is here necessary and useful. In fact, looking at the label *disease tissue* that considers the cancer types inside each tissue the result is very encouraging. In fact the score is quite high and the results promising. To gain better scores in this situation where samples are affected by the cancer complexity and heterogeneity is probably necessary to add more genes to the network.

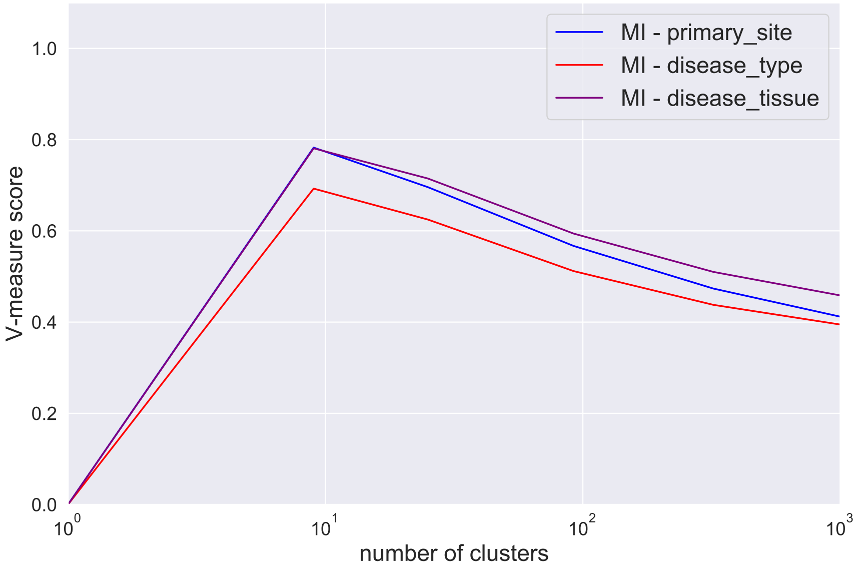


Figure 5.25: Score across the hierarchy for TCGA. In blue the primary site labels were considered, in red the disease types and in purple the mix of the two.

Looking directly into the cluster composition the tissue separation is quite good and visually appreciable. In figure 5.26 clusters at the higher level of the hierarchy. Some tissues are well separated at this point, at the same time the model seems to group the samples by system: digestive system is the more evident. Going deeper in

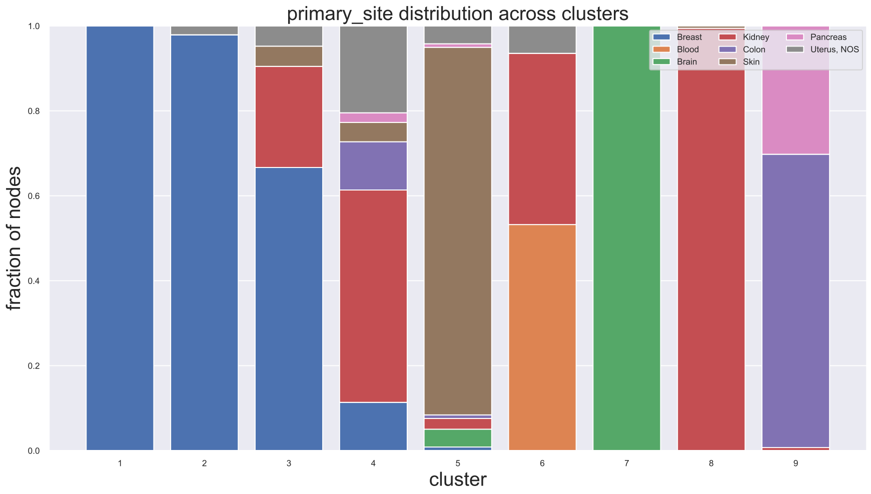


Figure 5.26: Clusters of diseased tissues at the higher level of the hierarchy. Breast is well separated, such as skin and brain. Cluster 9 contains digestive systems samples from pancreas and colon.

the hierarchy the tissue separation becomes visually appreciable and all the clusters are almost tissue-specific. This is clear in figure 5.27 which shows the primary site of the tumours well classified.

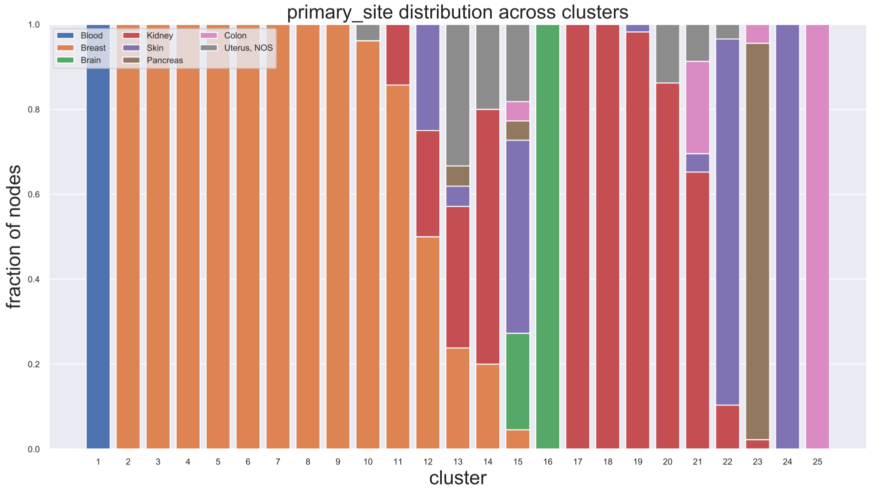


Figure 5.27: Normalized cluster composition from diseased samples. The primary site is here reported.

Going further, deep in the hierarchy, the disease type associated with each site emerges. In figure 5.28 it is shown that each tissue is then separated between different disease types. Note that the pure disease type classification is not useful since certain types of tumours can appear in different sites. In this case, the power of a hierarchic approach is evident: firstly the sites are retrieved, but going further also the disease type is classified quite well.

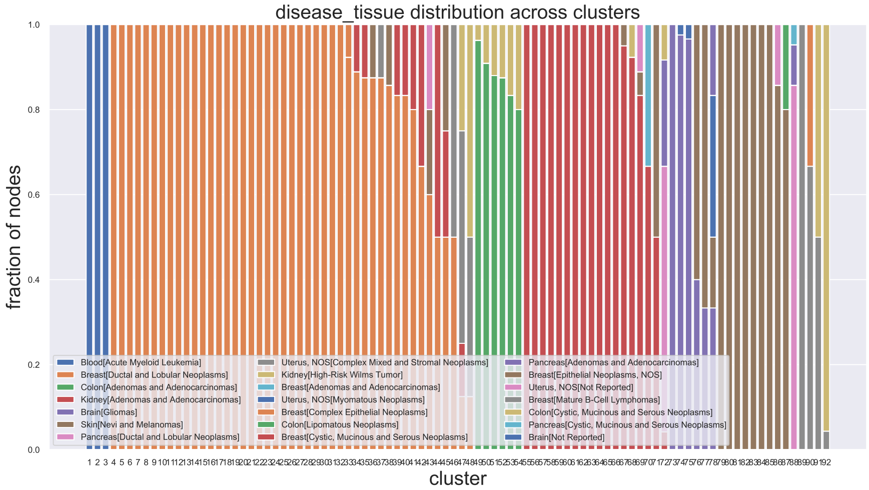


Figure 5.28: Normalized cluster composition of diseased tissue or couple site and disease type.

At this point when the model is demonstrated to work on healthy and diseased samples, it can be interesting to study merged healthy and diseased labels and examine how the model behaves when healthy and cancer samples are merged. It can be very useful to determine when the model identifies a diseased sample and when it is able to classify it properly.

### 5.3.6 Healthy and diseased together

In the previous sections it was demonstrated that the model works on samples from different datasets and performs well on both healthy and diseased samples. It can be interesting to see how the model behaves when both kinds of data are presented to it. The goal of this part of work is to identify which genes or topics identify and distinguish tissues themselves and which drive cancer and are necessary to understand the differentiation between cancer types.

For this analysis data from GTEx and TCGA were still analysed, but from a particular dataset available from [20] were authors tried to unify the normalization process from different dataset and sources [21]. This is, in practice, a mixed bigger dataset; note that not every tissue is present in both GTEx and TCGA, so only common tissues are considered here. The first label considered at this point is the tissue primary site, forgetting about its status (healthy or diseased), the secondary label refers to the tissues but separates their status. For example, a healthy brain sample from GTEx and a cancer brain from TCGA share the *brain* primary site label but have different secondary site assignments.

Once the model is run, the first element to look at is the V-measure; in figure 5.29 the result for the primary site is quite satisfying: clusters are very homogeneous and V-measure's peak is near 0.8.

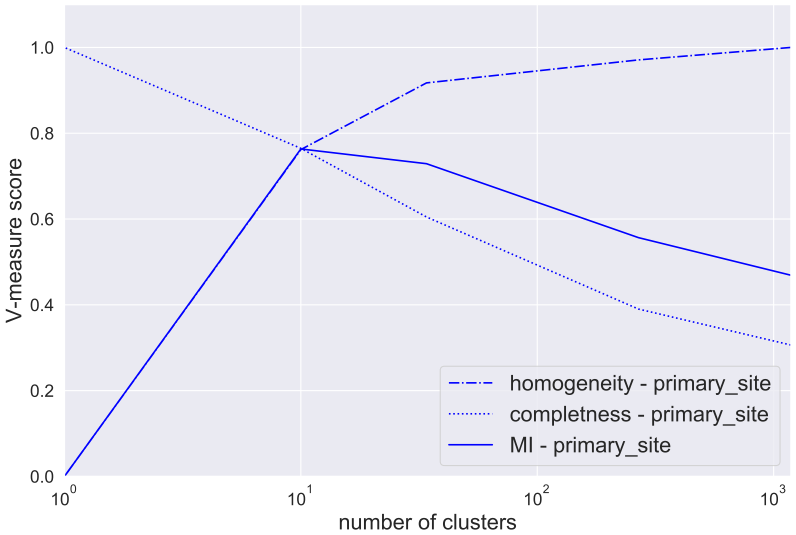


Figure 5.29: V-measure score for the run with merged healthy and diseased samples. Homogeneity, completeness and mutual information are represented.

Estimating the score also for the secondary site, or rather for the tissues with the health state and just the healthy/disease label lead to figure 5.30. This result is quite interesting, first of all even the secondary label is well classified and this happens at deeper level with respect to the one where tissues are separated; this means that firstly samples are separated by tissues then by their health state. This is very interesting because it is an evidence that the model recognize tissues, never mind where they come from, moreover the difference between datasets are not important here and so the normalization made by [21] brings no problems at this level. Moreover, looking just at the health status label the score is quite low (below 0.2) so the model does not take over the difference between datasets. To conclude the score analysis a mixed score

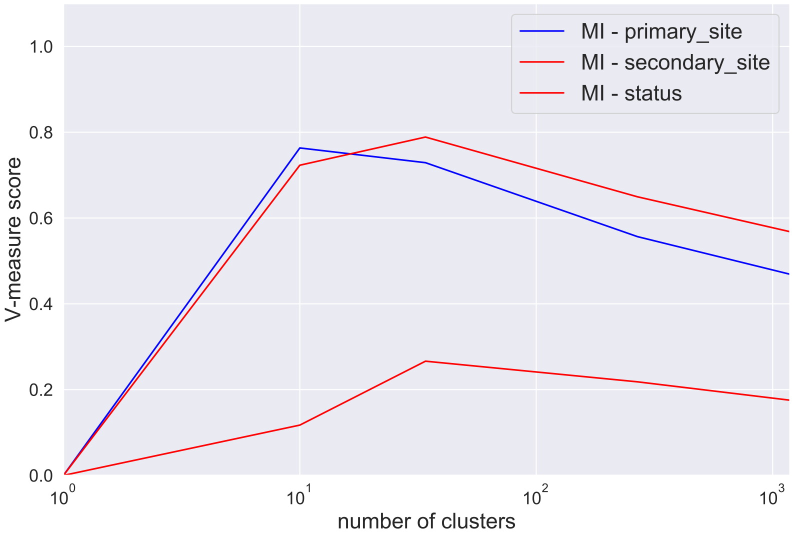


Figure 5.30: V-measure score for the run with merged healthy and diseased samples. Primary site (brain, blood, pancreas. . . ) labels are compared with secondary labels (healthy brain, brain cancer, healthy blood, blood cancer, healthy pancreas, pancreas cancer. . . ). The health status label (healthy / score diseased) is plotted.

is considered (the homogeneity of the primary site is considered with the completeness of secondary label) so that the score increases if going deeper in the hierarchy the separation of a homogeneous cluster brings to the separation of the refined labels. In figure 5.31 this score is compared with the ones obtained with LDA, hierarchical clustering and the null model. What happened here is that hierarchical Stochastic Block Model performs the best, LDA approach is good, hierarchical clustering has a quite bad score and all are better than the reshuffling null model.

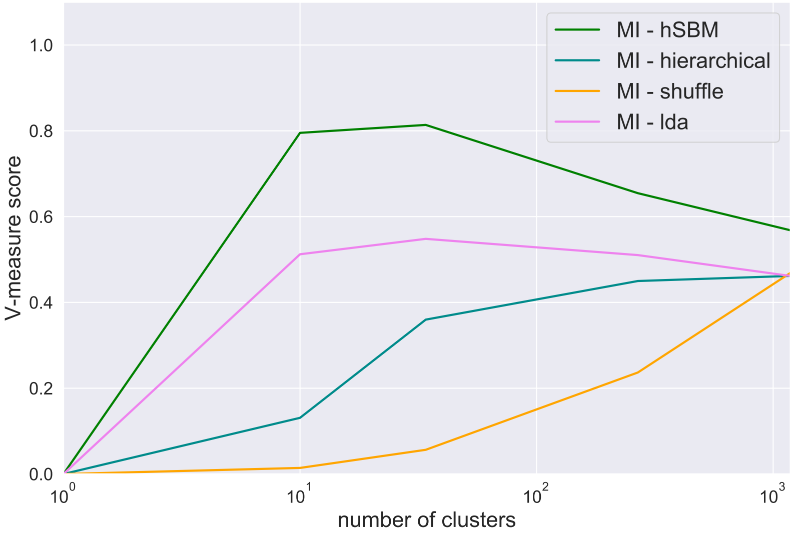


Figure 5.31: V-measure score for run with merged healthy and diseased samples. LDA, hierarchical clustering and null model for comparison.

**Gene sets** analysis is then performed. Considering the P-value of the term which P-value was the lowest, one P-value for each topic at the level of the hierarchy where the V-measure was maximized, it is possible to realize the $-Log_{10}(\mathrm{P-value})$ histogram. The tests are quite interesting, in fact there is an enrichment with a P-value lower than 0.05 in most cases, so it is possible to assert that topics carry some interesting information more than expected by picking up genes at random. In figure 5.33 the categories of the terms with lower P-values are shown. This explains what aspect of the samples a topic describes. The majority of terms found in topics comes from

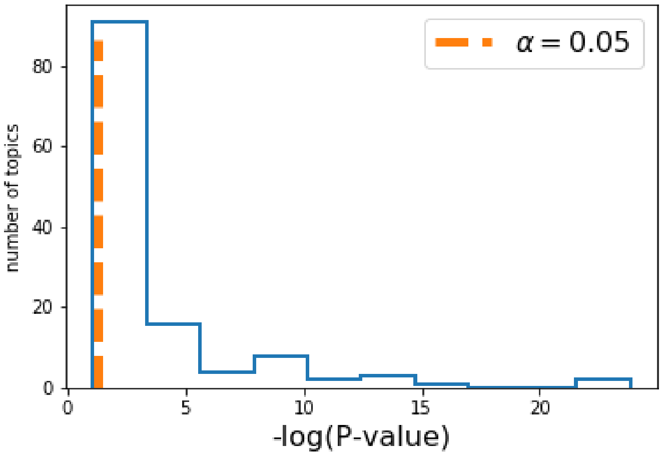


Figure 5.32: $-Log_{10}(\mathrm{P-value})$ of the term with the lowest P-value in each topic. In orange the standard 0.05 threshold.

the GTEx annotation for tissue expression, many are from GO biological process, GO molecular function and some from Human phenotype ontology. Nevertheless, some topics present enrichment for *NCI-60 Cancer Cell Lines*, meaning that these topics contains genes that are somehow cancer-related.

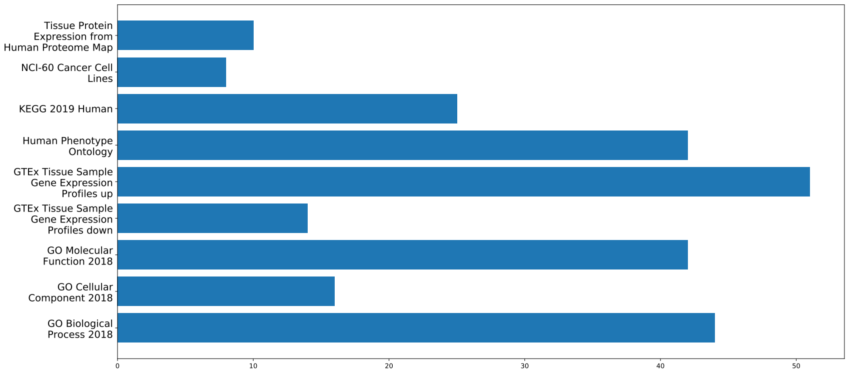
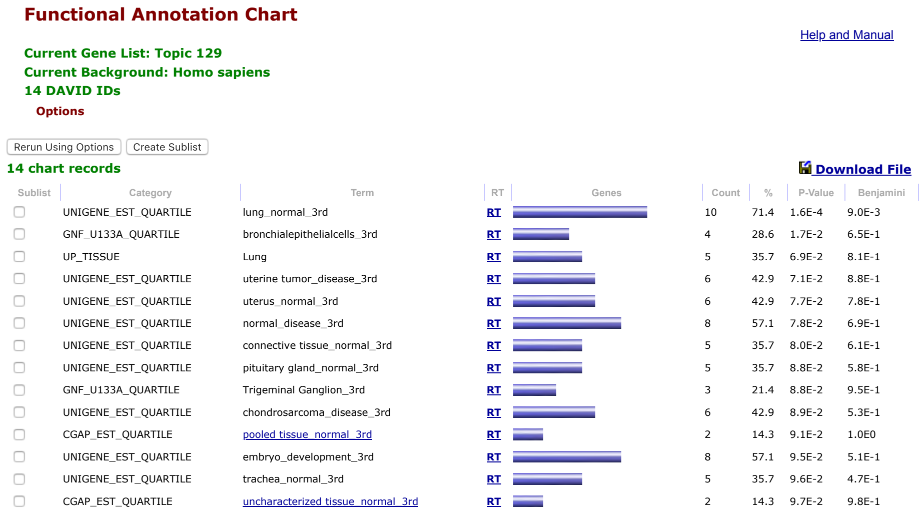


Figure 5.33: Categories of the terms with lower P-values in each topic.

Going forward in the analysis it is possible to perform enrichment test with other tools such as DAVID [39, 40]. Results are similar to the ones retrieved before. Tissue-related terms are found also using this tool; this confirms the absence of tool, sets or categories related biases. In figures 5.34, 5.35 and 5.36 the result from DAVID enrichment analysis. Finally, it is interesting to notice that topics are quite small (order $\simeq 20$ genes), so there aren't biases that can appear doing enrichment tests on big sets.



**Functional Annotation Chart**

Help and Manual

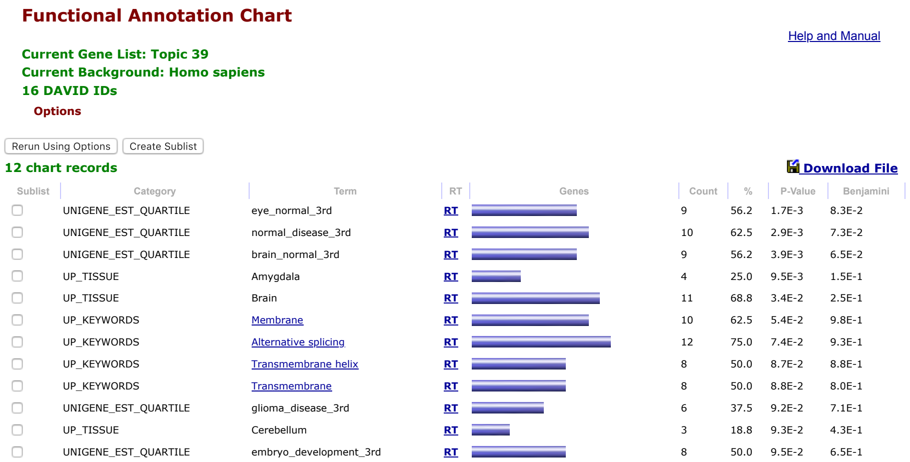
**Current Gene List: Topic 129**
**Current Background: Homo sapiens**
**14 DAVID IDs**
**Options**

**14 chart records**

Download File

| Sublist | Category | Term | RT | Genes | Count | % | P-Value | Benjamini |
|---|---|---|---|---|---|---|---|---|
| □ | UNIGENE_EST_QUARTILE | lung_normal_3rd | RT | | 10 | 71.4 | 1.6E-4 | 9.0E-3 |
| □ | GNF_U133A_QUARTILE | bronchialepithelialcells_3rd | RT | | 4 | 28.6 | 1.7E-2 | 6.5E-1 |
| □ | UP_TISSUE | Lung | RT | | 5 | 35.7 | 6.9E-2 | 8.1E-1 |
| □ | UNIGENE_EST_QUARTILE | uterine tumor_disease_3rd | RT | | 6 | 42.9 | 7.1E-2 | 8.8E-1 |
| □ | UNIGENE_EST_QUARTILE | uterus_normal_3rd | RT | | 6 | 42.9 | 7.7E-2 | 7.8E-1 |
| □ | UNIGENE_EST_QUARTILE | normal_disease_3rd | RT | | 8 | 57.1 | 7.8E-2 | 6.9E-1 |
| □ | UNIGENE_EST_QUARTILE | connective tissue_normal_3rd | RT | | 5 | 35.7 | 8.0E-2 | 6.1E-1 |
| □ | UNIGENE_EST_QUARTILE | pituitary gland_normal_3rd | RT | | 5 | 35.7 | 8.8E-2 | 5.8E-1 |
| □ | GNF_U133A_QUARTILE | Trigeminal Ganglion_3rd | RT | | 3 | 21.4 | 8.8E-2 | 9.5E-1 |
| □ | UNIGENE_EST_QUARTILE | chondrosarcoma_disease_3rd | RT | | 6 | 42.9 | 8.9E-2 | 5.3E-1 |
| □ | CGAP_EST_QUARTILE | pooled tissue_normal_3rd | RT | | 2 | 14.3 | 9.1E-2 | 1.0E0 |
| □ | UNIGENE_EST_QUARTILE | embryo_development_3rd | RT | | 8 | 57.1 | 9.5E-2 | 5.1E-1 |
| □ | UNIGENE_EST_QUARTILE | trachea_normal_3rd | RT | | 5 | 35.7 | 9.6E-2 | 4.7E-1 |
| □ | CGAP_EST_QUARTILE | uncharacterized tissue_normal_3rd | RT | | 2 | 14.3 | 9.7E-2 | 9.8E-1 |

Figure 5.34: Enrichment test on DAVID platform reveals lung-related genes.



**Functional Annotation Chart**

Help and Manual

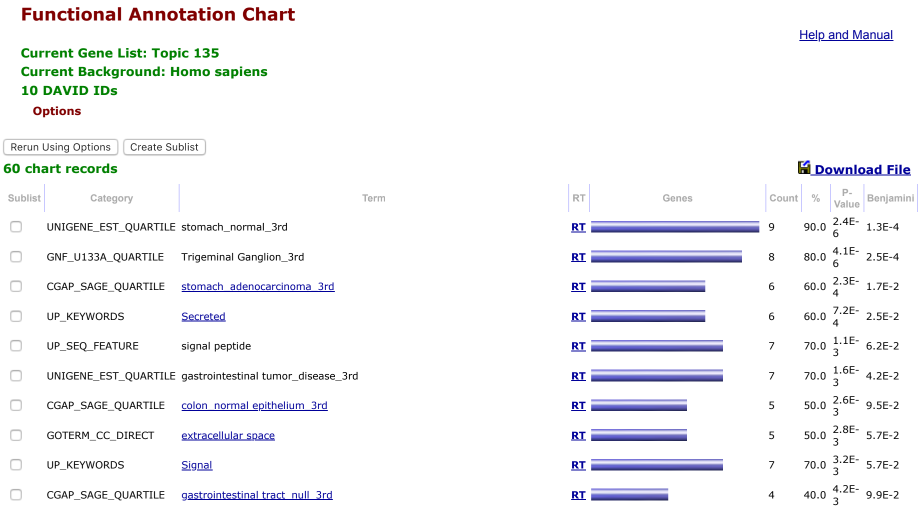
**Current Gene List: Topic 39**
**Current Background: Homo sapiens**
**16 DAVID IDs**

**Options**

**12 chart records**

| Sublist | Category | Term | RT | Genes | Count | % | P-Value | Benjamini |
|---|---|---|---|---|---|---|---|---|
| □ | UNIGENE_EST_QUARTILE | eye_normal_3rd | RT | | 9 | 56.2 | 1.7E-3 | 8.3E-2 |
| □ | UNIGENE_EST_QUARTILE | normal_disease_3rd | RT | | 10 | 62.5 | 2.9E-3 | 7.3E-2 |
| □ | UNIGENE_EST_QUARTILE | brain_normal_3rd | RT | | 9 | 56.2 | 3.9E-3 | 6.5E-2 |
| □ | UP_TISSUE | Amygdala | RT | | 4 | 25.0 | 9.5E-3 | 1.5E-1 |
| □ | UP_TISSUE | Brain | RT | | 11 | 68.8 | 3.4E-2 | 2.5E-1 |
| □ | UP_KEYWORDS | Membrane | RT | | 10 | 62.5 | 5.4E-2 | 9.8E-1 |
| □ | UP_KEYWORDS | Alternative splicing | RT | | 12 | 75.0 | 7.4E-2 | 9.3E-1 |
| □ | UP_KEYWORDS | Transmembrane helix | RT | | 8 | 50.0 | 8.7E-2 | 8.8E-1 |
| □ | UP_KEYWORDS | Transmembrane | RT | | 8 | 50.0 | 8.8E-2 | 8.0E-1 |
| □ | UNIGENE_EST_QUARTILE | glioma_disease_3rd | RT | | 6 | 37.5 | 9.2E-2 | 7.1E-1 |
| □ | UP_TISSUE | Cerebellum | RT | | 3 | 18.8 | 9.3E-2 | 4.3E-1 |
| □ | UNIGENE_EST_QUARTILE | embryo_development_3rd | RT | | 8 | 50.0 | 9.5E-2 | 6.5E-1 |

Figure 5.35: Enrichment test on DAVID platform reveals brain-related genes.



**Functional Annotation Chart**

Help and Manual

**Current Gene List: Topic 135**
**Current Background: Homo sapiens**
**10 DAVID IDs**

**Options**

**60 chart records**

| Sublist | Category | Term | RT | Genes | Count | % | P-Value | Benjamini |
|---|---|---|---|---|---|---|---|---|
| □ | UNIGENE_EST_QUARTILE | stomach_normal_3rd | RT | | 9 | 90.0 | 2.4E-6 | 1.3E-4 |
| □ | GNF_U133A_QUARTILE | Trigeminal Ganglion_3rd | RT | | 8 | 80.0 | 4.1E-6 | 2.5E-4 |
| □ | CGAP_SAGE_QUARTILE | stomach_adenocarcinoma_3rd | RT | | 6 | 60.0 | 2.3E-4 | 1.7E-2 |
| □ | UP_KEYWORDS | Secreted | RT | | 6 | 60.0 | 7.2E-4 | 2.5E-2 |
| □ | UP_SEQ_FEATURE | signal peptide | RT | | 7 | 70.0 | 1.1E-3 | 6.2E-2 |
| □ | UNIGENE_EST_QUARTILE | gastrointestinal tumor_disease_3rd | RT | | 7 | 70.0 | 1.6E-3 | 4.2E-2 |
| □ | CGAP_SAGE_QUARTILE | colon_normal epithelium_3rd | RT | | 5 | 50.0 | 2.6E-3 | 9.5E-2 |
| □ | GOTERM_CC_DIRECT | extracellular space | RT | | 5 | 50.0 | 2.8E-3 | 5.7E-2 |
| □ | UP_KEYWORDS | Signal | RT | | 7 | 70.0 | 3.2E-3 | 5.7E-2 |
| □ | CGAP_SAGE_QUARTILE | gastrointestinal tract_null_3rd | RT | | 4 | 40.0 | 4.2E-3 | 9.9E-2 |

Figure 5.36: Enrichment test on DAVID platform reveals stomach-related genes.

**The link between topics and samples** has not been investigated so far. The probability distribution of each sample over topics $P(\text{topic}|\text{sample})$ can be estimated as

$$P(\text{topic}|\text{sample}) = \frac{\#\text{ of edges on sample from topic}}{\#\text{ of edges on sample}}$$

after the model is run. Moreover, an average of all samples belonging to a topic can be estimated: $P(\text{topic}|\text{tissue}) = \frac{1}{|tissue|}\sum_{sample \in tissue} P(\text{topic}|\text{sample})$.

In figure 5.37 $P(\text{topic}|\text{tissue})$ is plotted for the first topics. What is clear is that in all samples there is a global trend without many differences between tissues, the topic expression differences between tissues are slightly appreciable at this point. This new point of view carries a profound and very informative message: in nature every tissue needs somehow the expression of all the genes (there is a global trend) and small differences between genes' expression are fine-tuned to obtain different tissues. In other words, it is possible to describe human tissues assuming that all genes are important and that is the fine structure of their interactions which realizes the complexity observed. In the case of diseased samples, this suggests that it should be possible to discover a cancer type not looking at a few marker genes but looking at the whole expression profile of all the genes.

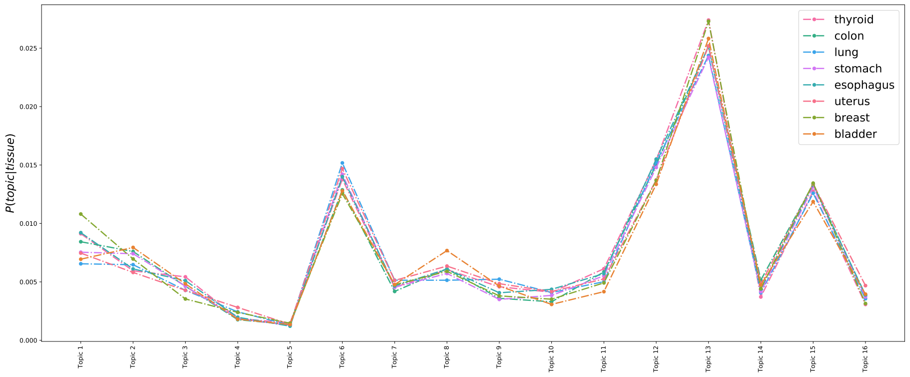


Figure 5.37: $P(\text{topic}|\text{tissue})$ for some topic coloured by tissue. It is evident a global trend and in some topics there are little differences between tissues.

In order to better understand these differences between topic expression in different tissues some kind of normalization inside each topic is needed. Here it was chosen to study inside each topic which tissues are most differently expressed (in average). To do so from each $P(\text{topic}|\text{tissue})$ it was subtracted the average topic expression $\text{mean}_{\text{tissue}}(\text{topic}) = \langle P(\text{topic}|\text{tissue})\rangle_{tissue}$ and the result was divided by the standard deviation $\sigma_{\text{tissue}}(\text{topic})$. In figure 5.38 some most characteristic topics are reported. This analysis reveals that different tissues have a different distance from average in different topics. When a tissue is distant from the average in a topic, usually that topic means something for that particular tissue. This analysis is useful to determine what is the role of each topic, moreover if a topic reveals cancer the difference between the healthy tissue and its diseased counterpart emerges as shown in the figure.

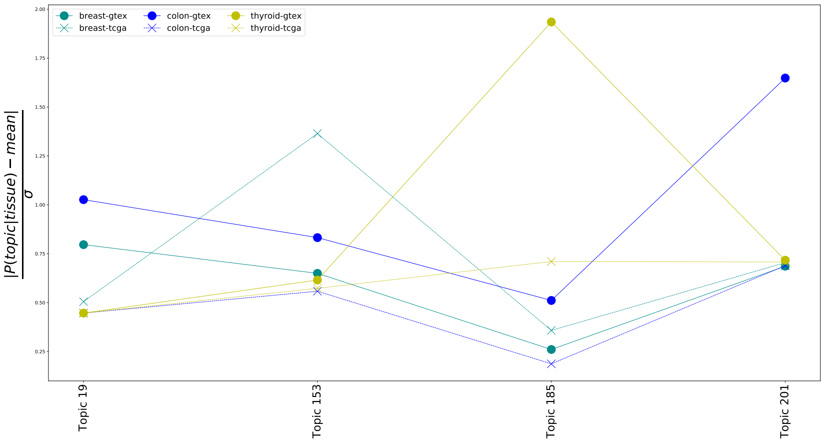


Figure 5.38: $\frac{\left|P(topic|tissue) - \langle P(\text{topic}|\text{tissue})\rangle_{tissue}\right|}{\sigma_{\text{tissue}}(\text{topic})}$ or the distance of each tissue from the average tissue expression in each topic. Some low occurrence topics are reported. Note *breast* cancer and healthy *thyroid* emerge.

The study of the relationship between topics and samples concludes the topic modelling analysis. In the next section, all results achieved will be summarized.

## 5.4 Results

The analysis using topic modelling leads to many interesting results.

The first result achieved is the development of a model that can reproduce the distinction between different tissues from RNA-sequencing datasets. This is evident looking at the cluster composition. Moreover, if one defines more objective metrics, based on entropy definition, the score is quite high, this encouraged further analysis. In particular, in many cases, the model not only reproduces the main tissue classification but was demonstrated that at different layers of the hierarchy even the sub-tissue labels are distinguished. The normalized mutual information score confirms this model's behaviour: tissues are separated at a higher level of the hierarchy and in the deeper layers the sub-tissues are distinguished.

A null model realized shuffling the labels confirms that the results achieved are non-trivial. Studying some quantities such as the fraction of cluster with the same label or the number of labels in a cluster and comparing these with the null models' ones it is possible to affirm that clusters are more homogeneous than expected.

The output of the model presented in this work (hSBM) was compared with standard approaches such as Latent Dirichlet Allocation and hierarchical clustering. hSBM outputs better results than standard approaches, moreover, it gains higher scores. An interesting fact is that topic modelling (both hSBM and LDA) is better than standard algorithms. This confirms the good quality of a topic model approach and, inside topic models, hSBM seems better than LDA. All algorithms are distant from the null model as highly expected.

It was also demonstrated that not only clustering of the sample was satisfying but even the genes' classification is interesting. If one looks at the block of genes, the so-called topics, enrichment tests confirm that topics represent an interesting group of genes. In particular, some dataset-specific labels were found in GTEx analysis.

In the end, the relationship between samples and topics reveals interesting facts. The distribution of the topic abundance across samples reveals that it is possible to describe the tissue differentiation as a complex mechanism of relationships between genes' expression. This isn't possible using an LDA approach where genes can have either a uniform or peaked distribution. Biologically this means that all genes are necessary everywhere and a fine-tuning of their expression differentiate by tissue.

The sample clustering, topic analysis and the relationship between samples and topics were made on three different datasets. GTEx containing just healthy samples, TCGA containing cancer samples and [20] which merged the two. Analysing the dataset with both healthy and diseased samples the differentiation between tissues is still evident and going deeper in the hierarchy the separation involves also the healthy or diseased status. The tissue separation in the firsts layer confirms that what the algorithm does is separating tissues and there isn't a bias involving differences between datasets.

Each of these cases reveals interesting facts. Being able to reproduce GTEx labels is a good benchmark of the algorithm performance; reproducing TCGA labels accurately is the real challenge that can improve scientific community knowledge and here was achieved, at least at primary-site level. Analysing both at the same time helps in understanding which genes are somehow involved in cancer development.

# Chapter 6

# Conclusions

Finally, this work demonstrates that RNA-sequencing datasets can be analysed from a component systems point of view.

Gene expression data show typical trends well-known, for example, in linguistics, moreover some interesting biological signatures were found. RNA-sequencing datasets have a great core of protein-coding genes that express everywhere, this is evident looking at $U$s or at Heaps' law. The presence of a power law distribution in the ranked abundances, the so-called Zipf's law is observed and characterizes the distribution of genes expression data.

In the first part of this work, a dataset (GTEx) containing samples from healthy tissues was analysed. One of the most interesting evidence was the presence of different Zipf's law when considering each tissue independently. Very similar results were obtained considering TCGA, a dataset containing thousands of cancer samples.

The power law distributions and the similarities with text analysis encouraged to explore the possibility of using a topic model approach to reveal the hidden structure of these datasets. This approach, originally developed to classify text, was useful to find clusters of samples that share some properties and to find the relationship between the genes and the samples.

Many goals were achieved during the topic modelling analysis. The pipeline begins filtering the data and selecting highly variable genes, then this network is processed with hierarchical Stochastic Block Model algorithm, then clusters are analysed and enrichment tests are performed; in the meanwhile, an objective and a well-defined score is estimated. All the analysis confirmed that this approach is successful. Three different datasets were analysed and, in every case, the model performed well. What was found is that clusters contain samples that share some properties, in particular, the tissue they are related to; enrichment tests found tissue-related terms in the topics along with Gene Ontology terms. The relationship between genes and samples is non-trivial and revealed a complex structure in gene expression data. Nevertheless, this structure is sufficient to discriminate between tissues.

In conclusion, topic modelling reveals itself as a useful approach to find the hidden structure of gene expression data. The prior analysis to select highly variables genes makes it possible to run the algorithm faster without losing the necessary information.

During this work the foundations have been laid for further analyses; for example, it should be interesting to study the variability of gene expression between tissues and between individuals, ideally genes that change their behaviour inter different tissues and not intra the same tissue are more likely tissue-specific. Trying to remove the sampling effect from the data could be another interesting analysis, in particular reproducing [41] on RNA-sequencing data could lead to the removal of sampling effects, here this was done just considering the sampling as a $CV^2$ boundary. Derive analytically the expression of the bound could be another goal. Use an innovation dynamic point of view to study the matrices discussed in this work can lead to other interesting results. Applying the model to single cell RNA-sequencing data, maybe from other kinds of animal, will present new challenges not present in bulk RNA-sequencing datasets considered here. Reproduce mouse data from [42] could be an interesting starting point. Maybe it worth to run the model on null data where just sampling is present and verify if there is some bias on the model due to the presence of the sampling.

The main future development of this work is indeed to run the model on a specific cancer tissue and find cancer subtypes, for instance, the breast or colon-rectal ones. Obtaining a hierarchy that at some level is able to identify cancer sub-types would be an ideal and great goal surely able to push further the human knowledge about cancer.

# Appendix A

# Hierarchical stochastic block model

The algorithm called hierarchical Stochastic Block Model (hSBM) from [5] is hereby summed up.

The first step of hierarchical stochastic block model, as discussed in [43], consists of creating a bipartite network $G$ with two kinds of nodes: **words** and **documents**. Every time a word $w$ is present in a document $d$ an edge $e_{wd}$ is created. If a word count in the entire corpus is under a certain threshold, that word could be ignored (this is not done in this work).

This model belongs to the so-called *generative models*: given the data, the model should generate a network $G$ with probability $P(G|\theta, b)$, where $b$ is the partition and $\theta$ any additional parameter of the model. The aim is to find a partition $b \in \{b_i\}$ with $B = |\{b_i\}|$ blocks.

Using the well-known Bayes theorem one could estimate the probability that an observed network was generated by the partition $b$:

$$P(b, \theta|G) = \frac{P(G|b, \theta) \overbrace{P(b, \theta)}^{prior}}{\underbrace{P(G)}_{\sum_\theta P(G|\theta,b)P(\theta,b)}} \tag{A.1}$$

defining the amount of information needed to describe the data as the description length

$$\Sigma = -lnP(G|b, \theta) - lnP(b, \theta) \tag{A.2}$$

the Eq.(A.1) can be written as $\frac{e^{-\Sigma}}{P(G)}$, so maximizing the posterior probability is equivalent to minimize the description length in Eq.(A.2). The probability of obtaining a graph from a set of parameters is $P(G|b, \theta) = \frac{1}{\Omega(A,\{n_r\})}$, where $\Omega(A, \{n_r\})$ is the number of graphs that is possible to generate with adjacency matrix $A$ and $\{n_r\}$ the counts distribution of block partition $\{b_i\}$ sizes. Note that, given an ensemble of network states, it is possible to define an entropy:

$$\Sigma = S - lnP(\theta),$$

with $S = Ln\,(\Omega)$. Note that if the number of group increases $S$ decreases but the prior $P(\theta)$ increases and avoids the risk of over-fitting. So minimize $S$ one minimizes $\Sigma$ and maximize the posterior $P(\theta|G)$. In case of a weighted network the likelihood becomes $P(G, x|b, \theta)$, where $x$ are the weights.

**Algorithm**

First of all, a $B \times B$ matrix is created. The entry $e_{rs}$ of this matrix represents the number of links between nodes of group $r$ and nodes of group $s$, with $r, s \in \{b_i\}$. At the beginning, $B$ groups are formed at random and the initial $B$ is a hyper-parameter of the model.

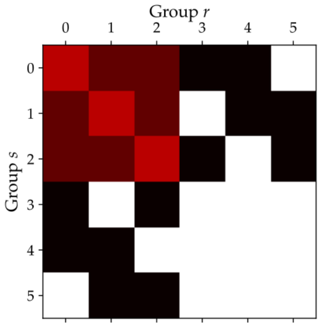


Figure A.1: Example of an edge's matrix from [13].

It is useful to define a traditional entropy:

$$S_t = \frac{1}{2}\Sigma_{r,s} n_r n_s H\left(\frac{e_{rs}}{n_r n_s}\right) \tag{A.3}$$

where $n_r$ is the number of nodes in groups $r$, $e_{rs}$ is the number of edges between nodes of group $r$ and nodes of group $s$, and $H(x) = -xlog_2(x) - (1-x)log_2(1-x)$. This entropy is equivalent to the micro-canonical entropy of a system with $\Omega(A, \{n_r\})$ accessible states $S_t = Ln\Omega$.

The algorithm uses a Markov Chain Monte Carlo to minimize this entropy. Made it simple, at each step a node changes block and the new configuration is accepted if $S$ is decreased.

Note that Eq.(A.3) can be corrected taking care of degree distribution obtaining corrected entropy $S_c$

$$S_c = -\Sigma_{r,s}\frac{e_{rs}}{2} - \Sigma_k N_k ln(k!) - \frac{1}{2}\Sigma_{r,s} e_{rs} ln\left(\frac{e_{rs}}{e_r e_s}\right), \tag{A.4}$$

being $k$ the degree distribution.

**How to change the group of a node?**

At each step according to [43] node $i$ can change group from $r$ to $s$ with a probability

$$P(r \to s|t) = \frac{e_{ts} + \epsilon}{e_t + \epsilon B} \tag{A.5}$$

where $j$ is a random neighbour of $i$: $j \in N_i$, $t \in \{b_j\}$ its block as defined in [43]. $\epsilon$ is a parameter that according to [32] hasn't significant impact in the algorithm, provided it is sufficiently small. Equation A.5 can be rewritten as

$$P(r \to s|t) = (1 - R_t)\frac{e_{ts}}{e_t} + \frac{R_t}{B}$$

defining $R_t = \frac{\epsilon B}{e_t + \epsilon B}$. The simulation consists of four steps: for each node $i$

- a node $j$ is chosen from $i$'s neighbours, the group of $j$ is called $t$;
- a random group $s$ is selected;
- move of node $i$ to group $s$ is accepted with probability $R_t$;
- if $s$ is not accepted, a random edge $e$ is chosen from group $t$ and node $i$ is assigned to the endpoint of $e$ which is not in $t$.

These steps mime probability in Eq.(A.5); note that for $\epsilon \to \infty$ this gives a uniform probability.

To enchant the probability to find a minimum, a bounce of these moves is made, only the set of moves with the minimum $S$ is accepted. Moreover, to remove eventual biases due to the initial configuration the model is run with different initial states, then the final state with the minimal entropy is selected.

### How many blocks $B$?

Note that the number of blocks $B$ is a free parameter and must be inferred as described in [32]. This implies a slight modification of the algorithm such that it can be possible to admit the creation of a new group. When a group $s$ is chosen, the algorithm can now accept a **new group** and Eq.(A.5) becames

$$P(r \to s) = \Sigma_t P(t|i)\frac{e_{ts} + \epsilon}{e_t + \epsilon(B+1)} \tag{A.6}$$

being $P(t|i) = \Sigma_j \frac{A_{ij}\delta(b_j,t)}{k_i}$ the fraction of neighbours of $i$ belonging to group $t$, $e_t$ the number of edges in group $t$, $k_i$ the degree, and $b_j$ groups.

Using this modification it is now possible to add new groups and $B$ is no longer a parameter.

### How to find hierarchical layers?

After the algorithm is run, one would add a new hierarchic level, this is done considering the $B$ groups as nodes and repeating the process. As done before a matrix of edges like figure A.1 is created, where edges are considered between groups of the previous layer. The posterior probability became

$$P(\{b_l\}|A) = \frac{P(A|\{b_l\})P(\{b_l\})}{P(A)} = \prod_l^L P(b_l|e_l, b_{l-1}) \tag{A.7}$$

where $l = 0 \ldots L$ is the layer, $A$ the audience matrix, $b_i$ blocks. Note that $e_0 = A$ and $B_L = 1$. Maximizing Eq.(A.7) gives the correct number of layers.

Adding a layer is done in 3 steps described in [44]:

Resize: find $B_l \in [B_{l-1}, B_{l+1}]$ by bisection;

Insert: a layer $l$;

Delete: $l$ and linking nodes from layer $l-1$ directly to groups of layer $l+1$.

One marks initially all levels as *not done* and starts at the top-level $l = L$ [44]. For the current level $l$, if it is marked *done* it is skipped and one moves to the level $l-1$. Otherwise, all three moves are attempted. If any of the moves succeed in decreasing the description length $\Sigma$ (Equation A.2), one marks the levels $l-1$ and $l+1$ (if they exist) as *not done*, the level $l$ as *done*, and one proceeds (if possible) to the upper level $l+1$, and repeats the procedure. If no improvement is possible, the level $l$ is marked as *done* and one proceeds to the lower level $l-1$. When the lowest level $l = 0$ is reached and cannot be improved, the algorithm ends.

**Overlapping partitions**

As described in [45] one of the advantages of this approach is that it is possible to let a node belonging to multiple groups. In this case $b_i$ becomes $\vec{b_i}$, with component $b_{ir} = 1$ if node $i$ is in group $r$, 0 otherwise. The number of 1s in vector $\vec{b_i}$ is called $d_i = |\vec{b_i}|$.

The probability of having a graph $G$ being generated from an adjacency matrix $A$ and a partition $\{\vec{b_i}\}$ is

$$P(G|A, \{\vec{b_i}\}) = \frac{1}{\Omega}$$

if $\Omega$ is the number of possible graphs. Entropy in Eq.(A.3) is $S_t = Ln\Omega$. This corresponds to an augmented graph generated via a non overlapping block model with $N' = \Sigma_r n_r > N$ nodes and the same adjacency matrix $A$.

First of all, it is necessary to sample the distribution of mixture sizes $P(\{n_d\})$ where $n_d$ is the number of nodes which mixture has got size $d$, $n_d \in [0, N]$ and $d \in [0, D]$ (typically $D = B$ and in the non-overlapping case $D = 1$), this is done by sampling uniformly from

$$P(\{n_d\}|B) = \left(\!\!\binom{D}{N}\!\!\right)^{-1}$$

which is probability of having $n$ nodes whose mixture has size $d$. $\left(\!\binom{B}{N}\!\right)$ is the number of histograms with area $N$ and $B$ distinguishable bins. $B-1$ can be used instead of $B$ to avoid node with no group, in this case $d \in [1, B]$.

Given the mixture sizes, the distribution of node membership is sampled from

$$P(\{d_i\}|\{n_d\}) = \frac{\prod_d n_d!}{N!}.$$

At this point for each set of nodes with $d_i = d$ it is necessary to sample $n_{\vec{b}}$: the number of nodes with a particular mixture $\vec{b}$. It is sampled from

$$P(\{n_{\vec{b}}\}_d|n_d) = \left(\!\!\binom{\binom{D}{d}}{n_d}\!\!\right)^{-1}, \tag{A.8}$$

next all mixtures $\vec{b_i}$ of size $d$ must be sampled, they are given by

$$P(\{\vec{b_i}\}_d|\{n_{\vec{b}}\}_d) = \frac{\prod_{|\vec{b_i}|=d} n_b!}{n_d!} \tag{A.9}$$

the global posterior as defined in [45] is

$$P(\{\vec{b_i}\}|B) = \left[\prod_{d=1}^{B} P(\{\vec{b_i}\}_d|\{n_{\vec{b}}\}_d)P(\{n_{\vec{b}}\}_d|n_d)\right] P(d_i|n_d)P(n_d|B) \tag{A.10}$$

At this time it is necessary to obtain the distribution of the edges between mixtures. Defined $e_r = \Sigma_s e_{rs}$ the number of half-edges labelled $r$, $m_r = \Sigma_{\vec{b}} b_r$ the number of mixtures containing group $r$ the algorithm samples the probability distribution of the edges count

$$P(\{e_{\vec{b}}\}|\{\vec{b_i}\},A) = \prod_r \left(\!\!\binom{m_r}{e_r}\!\!\right)^{-1}$$

and the labelled degree sequence $\{\vec{k_i}\}$ from

$$P(\{\vec{k_i}\}_{\vec{b}}|\{e_{\vec{b}}\},\{\vec{b_i}\}) = \frac{\prod_k n_k^{\vec{b}}!}{n_{\vec{b}}!}$$

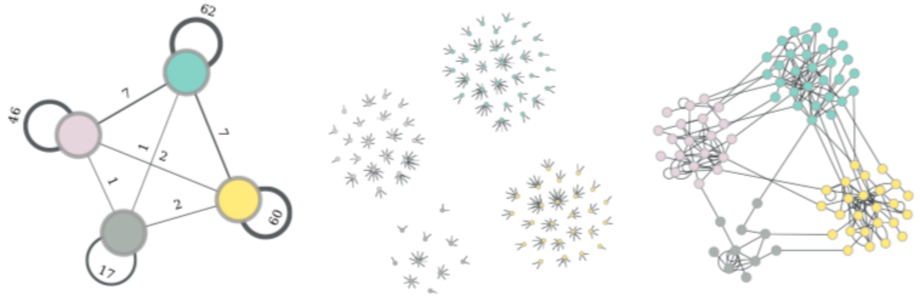


Figure A.2: Illustration of the generative process of the microcanonical SBM. Given a partition of the nodes, the edge counts between groups are sampled (left), followed by the degrees of the nodes (centre) and finally the network itself (right). From [30].

### Word documents separation

Following what is done in [5], the probability of a group $P(b_l)$ at a certain level $l$ is intended as the disjoint probability of group of words and group of documents:

$$P(b_l) = P_w(b_l^w)P_d(b_l^d). \tag{A.11}$$

Doing this let words and documents be separated by construction. Considering the process described above if two nodes are not connected at the beginning it is impossible that they end up in the same block. It is easily verified in [43] that this property is preserved and fully reflected in the final block structure.

# Appendix B

# Homogeneity, completeness and V-measure

Using algorithms that are unsupervised, but with a ground truth available it is useful to define some metrics.

One is the homogeneity

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{B.1}$$

defining the entropy

$$H(C|K) = \sum_{c \in \text{modellabels}, k \in \text{clusters}} \frac{n_{ck}}{N} Log\left(\frac{n_{ck}}{n_k}\right) \tag{B.2}$$

where $n_{ck}$ is the number of nodes of type $c$ in cluster $k$, $N$ the number of nodes and $n_k$ the number of nodes in cluster $k$. It is evident that if all nodes inside cluster $k$ are of the same type $c$ $n_{ck} = n_k$, $H(C|K) = 0$ and $h = 1$, it is actually a full homogeneous situation. The completeness:

$$c = 1 - \frac{H(K|C)}{H(K)}, \tag{B.3}$$

$H(K|C)$ is defined in the same way as $H(C|K)$. Completeness measures if all nodes of the same type are in the same cluster. Ideally one wants a model which output is both homogeneous and complete. So it is possible to define the V-measure [33], which is the harmonic average of the two:

$$2\frac{hc}{h+c}. \tag{B.4}$$

The product $hc$ is equal to

$$\frac{(H(C) - H(C|K))(H(K) - H(K|C))}{H(K)H(C)}, \tag{B.5}$$

the sum $h + c$ is

$$\frac{H(K)(H(C) - H(C|K)) + H(C)(H(K) - H(K|C))}{H(K)H(C)}. \tag{B.6}$$

Expressing the conditional entropy

$$H(K|C) = \sum_{kc} P(k,c)Log_2(P(k|c)) = \sum_{kc} P(k,c)Log_2\left(\frac{P(k,c)}{P(c)}\right) = H(K,C) - H(C)$$

in terms of the conjunct entropy $H(K,C)$ which is symmetric by exchanges of $C$ and $K$

$$H(K,C) = H(K|C) + H(C) = H(C|K) + H(K) = H(C,K)$$

it is easy to verify that

$$H(C) - H(C|K) = H(K) - H(K|C)$$

so

$$hc = \frac{(H(C) - H(C|K))^2}{H(K)H(C)}$$

and

$$h + c = \frac{(H(C) - H(C|K))(H(K) + H(C))}{H(K)H(C)}.$$

The harmonic average $2\frac{hc}{h+c}$ gives

$$\mathrm{V-measure} = 2\frac{H(C) - H(C|K)}{H(K) + H(C)} = 2\frac{H(C) + H(K) - H(K,C)}{H(K) + H(C)} = 2\frac{MI(C,K)}{H(K) + H(C)}$$

which is actually the mutual information between $P(C)$ and $P(K)$ normalized to 1 by the term $H(C) + H(K)$. In fact if $P(C) = P(K)$ $H(K,C) = H(K) = H(C)$ and the measure is 1, if $P(C)$ and $P(K)$ are completely independent $H(K,C) = H(K) + H(C)$ and the measure is 0.

# List of Figures

# List of Tables

# Bibliography

[1] TCGA Research Network, "Cell-of-Origin Patterns Dominate the Molecular Classification of 10,000 Tumors from 33 Types of Cancer," *Cell*, vol. 173, no. 2, pp. 291–304.e6, 2018.

[2] W. Zhou, S. Yao, L. Liu, L. Tang, and W. Dong, "An overview of topic modeling and its current applications in bioinformatics," *Springerplus*, 2016.

[3] A. Lancichinetti, M. Irmak Sirer, J. X. Wang, D. Acuna, K. Körding, and L. A. Amaral, "High-reproducibility and high-accuracy method for automated topic classification," *Phys. Rev. X*, 2015.

[4] A. Martini, A. Cardillo, and P. D. L. Rios, "Entropic selection of concepts unveils hidden topics in documents corpora," *arXiv*, 2017.

[5] M. Gerlach, T. P. Peixoto, and E. G. Altmann, "A network approach to topic models," *Science advances*, vol. 4, no. 7, p. eaaq1360, 2018.

[6] J. Siek, A. Lumsdaine, and L.-Q. Lee, *The boost graph library: user guide and reference manual.* Addison-Wesley, 2002.

[7] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, Austin, TX, 2010.

[8] E. Jones, T. Oliphant, and P. Peterson, "{SciPy}: Open source scientific tools for {Python}," *[Online; accessed 2019-07-12]*, 2014.

[9] T. E. Oliphant, *A guide to NumPy*, vol. 1. Trelgol Publishing USA, 2006.

[10] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in science & engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[11] M. A. et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[12] Z. et al., "Apache spark: A unified engine for big data processing," *Commun. ACM*, vol. 59, pp. 56–65, Oct. 2016.

[13] T. P. Peixoto, "The graph-tool python library," *figshare*, 2014.

[14] E. Fernández-del Castillo, D. Scardaci, and Á. L. García, "The egi federated cloud e-infrastructure," *Procedia Computer Science*, vol. 68, pp. 196–205, 2015.

[15] M. Aldinucci, S. Bagnasco, S. Lusso, P. Pasteris, S. Rabellino, and S. Vallero, "OCCAM: a flexible, multi-purpose and extendable HPC cluster," *Journal of Physics: Conference Series*, vol. 898, no. 8, p. 082039, 2017.

[16] Z. Wang, M. Gerstein, and M. Snyder, "Rna-seq: a revolutionary tool for transcriptomics," *Nature reviews genetics*, vol. 10, no. 1, p. 57, 2009.

[17] L. J. Carithers, K. Ardlie, *et al.*, "A novel approach to high-quality postmortem tissue procurement: the gtex project," *Biopreservation and biobanking*, vol. 13, no. 5, pp. 311–319, 2015.

[18] K. K. Dey, C. J. Hsiao, and M. Stephens, "Visualizing the structure of rna-seq expression data using grade of membership models," *PLoS genetics*, vol. 13, no. 3, p. e1006599, 2017.

[19] R. L. Grossman, A. P. Heath, V. Ferretti, H. E. Varmus, D. R. Lowy, W. A. Kibbe, and L. M. Staudt, "Toward a shared vision for cancer genomic data," *New England Journal of Medicine*, vol. 375, no. 12, pp. 1109–1112, 2016.

[20] Q. Wang, J. Gao, and N. Schultz, "Unified RNA-seq datasets in human cancers and normal tissues - normalized data," *figshare*, 2017.

[21] D. Betel, A. Ochoa, C. Zhang, A. V. Penson, L. Zhang, N. Schultz, C. A. Iacobuzio-Donahue, B. E. Gross, Q. Wang, J. Armenia, J. Gao, E. Reznik, T. Minet, and B. S. Taylor, "Unifying cancer and normal RNA sequencing data from different sources," *Sci. Data*, 2018.

[22] A. Mazzolini, A. Colliva, M. Caselle, and M. Osella, "Heaps' law, statistics of shared components, and temporal patterns from a sample-space-reducing process," *Physical Review E*, vol. 98, no. 5, p. 052139, 2018.

[23] A. Mazzolini, J. Grilli, E. De Lazzari, M. Osella, M. C. Lagomarsino, and M. Gherardi, "Zipf and Heaps laws from dependency structures in component systems," *Phys. Rev. E*, vol. 98, p. 012315, jul 2018.

[24] E. G. Altmann and M. Gerlach, "Statistical laws in linguistics," in *Creativity and Universality in Language*, pp. 7–26, Springer, 2016.

[25] A. Mazzolini, M. Gherardi, M. Caselle, M. Cosentino Lagomarsino, and M. Osella, "Statistics of Shared Components in Complex Component Systems," *Phys. Rev. X*, vol. 8, p. 021023, apr 2018.

[26] H. S. Heaps, *Information Retrieval: Computational and Theoretical Aspects.* Orlando, FL, USA: Academic Press, Inc., 1978.

[27] M. Melé, F. Pedro, R. Ferran, D. S. DeLuca, M. Jean, S. Micheal, K. Ardlie, and G. Roderic, "The human transcriptome across tissues and individuals," *Science (80-. ).*, vol. 348, no. 6235, pp. 660–665, 2014.

[28] Z. Eisler, I. Bartos, and J. Kertész, "Fluctuation scaling in complex systems: Taylor's law and beyond1," *Adv. Phys.*, vol. 57, pp. 89–142, jan 2008.

[29] S. Islam, A. Zeisel, S. Joost, G. La Manno, P. Zajac, M. Kasper, P. Lönnerberg, and S. Linnarsson, "Quantitative single-cell RNA-seq with unique molecular identifiers," *Nat. Methods*, vol. 11, p. 163, dec 2013.

[30] T. P. Peixoto, "Bayesian stochastic blockmodeling," *arXiv*, may 2017.

[31] P. W. Holland, K. Blackmond, and S. Leinhardt, "STOCHASTIC BLOCKMODELS: FIRST STEPS," tech. rep., CMU, 1983.

[32] T. P. Peixoto, "Nonparametric bayesian inference of the microcanonical stochastic block model," *Physical Review E*, vol. 95, no. 1, p. 012317, 2017.

[33] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[34] H. Shi, M. Gerlach, I. Diersen, D. Downey, and L. A. N. Amaral, "A new evaluation framework for topic modeling algorithms based on synthetic corpora," tech. rep.

[35] L. Zappia and A. Oshlack, "Clustering trees: a visualization for evaluating clusterings at multiple resolutions," *GigaScience*, vol. 7, jul 2018.

[36] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, *et al.*, "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 43, pp. 15545–15550, 2005.

[37] M. V. Kuleshov, M. R. Jones, A. D. Rouillard, N. F. Fernandez, Q. Duan, Z. Wang, S. Koplev, S. L. Jenkins, K. M. Jagodnik, A. Lachmann, M. G. McDermott, C. D. Monteiro, G. W. Gundersen, and A. Ma'ayan, "Enrichr: a comprehensive gene set enrichment analysis web server 2016 update.," *Nucleic Acids Res.*, vol. 44, no. W1, pp. W90–7, 2016.

[38] A. et al., "The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans," *Science (80-. ).*, vol. 348, pp. 648–660, may 2015.

[39] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists," *Nucleic acids research*, vol. 37, no. 1, pp. 1–13, 2008.

[40] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Systematic and integrative analysis of large gene lists using david bioinformatics resources," *Nature protocols*, vol. 4, no. 1, p. 44, 2009.

[41] J. Grilli, "Multiplicative growth model explains macroecological patterns across microbiomes — Supplementary Material —," -, pp. 1–15, 2019.

[42] A. Scialdone, Y. Tanaka, W. Jawaid, V. Moignard, N. K. Wilson, I. C. Macaulay, J. C. Marioni, and B. Göttgens, "Resolving early mesoderm diversification through single-cell expression profiling," *Nature*, vol. 535, p. 289, jul 2016.

[43] T. P. Peixoto, "Efficient monte carlo and greedy heuristic for the inference of stochastic block models," *Physical Review E*, vol. 89, no. 1, p. 012804, 2014.

[44] T. P. Peixoto, "Hierarchical block structures and high-resolution model selection in large networks," *Phys. Rev. X*, vol. 4, no. 1, 2014.

[45] T. P. Peixoto, "Model selection and hypothesis testing for large-scale network models with overlapping groups," *Physical Review X*, vol. 5, no. 1, p. 011033, 2015.

# Acknowledgements

Ringrazio per avermi permesso di fare questi cinque anni di Università prima di tutto i miei genitori Giorgio e Paola. Ringrazio Antonella per esserci sempre. I miei nonni Lina e Battista. Mia sorella Lucia e tutta la mia famiglia.

Mario che, con Antonella e Nicolò, mi ha fatto leggere Feynman e permesso di usare supercomputer ben prima che mi iscrivessi a Fisica.

Un saluto e ringraziamento ai miei amici senza i quali questi anni sarebbero stati sicuramente più faticosi: Simone, il primo che ho incontrato a Fisica; Diana e Giulia, amiche a distanza; Emanuele, Flavio, Federico, Marco, Lorenzo, Tiziano, Elisa con cui ho condiviso inenarrabili viaggi sulla canavesana durante la stesura di questo lavoro.

Dott. Ric. Gabriele, co-relatore, amico e socio i cui pensa-pensa il venerdì pomeriggio e i vari progetti continueranno.

Un pensiero speciale ad Anna senza la quale non sarei arrivato alla fine di questi lunghi 5 anni (probabilmente neanche dei primi 3). Cinque anni in cui siamo passati dal ricevimento di GAL I a stare insieme con infinite avventure che, spero, siano solo l'inizio di qualcosa di grande.

I Prof. Chiosso, Prof. Beolè, Prof. Bellan e Prof. Argirò la cui assistenza a Esp. II mi mancherà moltissimo.

Infine ringrazio l'intero gruppo ByoPhys
http://personalpages.to.infn.it/~caselle/BioPhys/BioPhys.html: Michele Prof. Caselle e Matteo Dott. Osella per avermi instradato a fare questo lavoro e per avermi supportato anche molto oltre le loro aree di competenza. Matteo Dott. Cereda per gli utili appunti durante tutto il lavoro. Francesco, Marta, Mattia, Eleonora, Marco, Serena, Gabriele è stato bello lavorare con voi, ci vediamo al PhD!